Probabilistic Conceptual Explainers: Trustworthy Conceptual Explanations for Vision Foundation Models

Hengyi Wang [*1] Shiwei Tan [*1] Hao Wang [1]

Abstract

Vision transformers (ViTs) have emerged as a significant area of focus, particularly for their capacity to be jointly trained with large language models and to serve as robust vision foundation models. Yet, the development of trustworthy explanation methods for ViTs has lagged, particularly in the context of post-hoc interpretations of ViT predictions. Existing sub-image selection approaches, such as feature-attribution and conceptual models, fall short in this regard. This paper proposes five desiderata for explaining ViTs – faithfulness, stability, sparsity, multi-level structure, and parsimony – and demonstrates the inadequacy of current methods in meeting these criteria comprehensively. We introduce a variational Bayesian explanation framework, dubbed ProbAbilistic Concept Explainers (PACE), which models the distributions of patch embeddings to provide trustworthy post-hoc conceptual explanations. Our qualitative analysis reveals the distributions of patch-level concepts, elucidating the effectiveness of ViTs by modeling the joint distribution of patch embeddings and ViT's predictions. Moreover, these patch-level explanations bridge the gap between image-level and dataset-level explanations, thus completing the multi-level structure of PACE. Through extensive experiments on both synthetic and real-world datasets, we demonstrate that PACE surpasses state-of-the-art methods in terms of the defined desiderata[1].

1. Introduction

Vision Transformers (ViTs) (Dosovitskiy et al., 2020) and their variants (Liu et al., 2021; Touvron et al., 2021; Radford et al., 2021) have emerged as pivotal models in computer vision, leveraging stacked self-attention blocks to encode raw inputs and produce patch-wise embeddings as contextual representations. Given their increasing application in high-risk domains such as autonomous driving, explainability has become a critical concern.

To date, post-hoc explanations in computer vision often involve attributing predictions to specific image regions. However, we identify two primary limitations in current methods: (1) Existing conceptual explanation methods (Fel et al., 2023a;b; Ghorbani et al., 2019; Zhang et al., 2021; Wang et al., 2020; Novello et al.; Chen et al., 2024; Li et al., 2021a; Wang et al., 2023) are not fully compatible with transformer-based models like vision transformers (ViTs), and they also fall short in offering a cohesive structure for dataset-image-patch analysis of input images. (2) Current state-of-the-art methods (Li et al., 2020; Pan et al., 2021; Agarwal et al., 2022; Colin et al., 2022; Xie et al., 2022; Wang et al., 2022; Fel et al., 2023b; Chen et al., 2023) evaluate visual concepts through subjective human utility scores or limited quantitative analysis, lacking a fair and consistent comparison framework. To address this, we propose a comprehensive set of desiderata for post-hoc conceptual explanations for ViTs, namely (see formal definitions in Sec. 3.2):

- *Faithfulness:* The explanation should be faithful to the explained ViT and able to recover its prediction.
- *Stability:* The explanation should be stable for different perturbed versions of the same image.
- *Sparsity:* For each prediction's explanation, only a small subset of concepts are relevant.
- *Multi-Level Structure:* There should be dataset-level, image-level, and patch-level explanations.
- *Parsimony:* There are a small number of concepts in total (see Appendix B for more details).

While previous research (Kim et al., 2018; Fel et al., 2023b; Oikarinen et al., 2023; Gilpin et al., 2018; Murdoch et al., 2019) has proposed and met different dimensions of the learned concepts, these studies often lack a comprehensive

[*]Equal contribution [1]Department of Computer Science, Rutgers University, New Jersey, USA. Correspondence to: Hengyi Wang <hengyi.wang@rutgers.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

[1]Code will soon be available at https://github.com/Wang-ML-Lab/interpretable-foundation-models

evaluation. In this paper, propose ProbAbilistic Concept Explainers (PACE) to provide trustworthy conceptual explanations aligned with these desiderata, drawing inspiration from hierarchical Bayesian deep learning (Wang & Yeung, 2016; 2020; Wang et al., 2016). For example, to enable ***multi-level*** explanations, we (1) model K concepts as a mixture of K Gaussian patch-embedding distributions, (2) treat the explained ViT's patch-level embeddings as observed variables, (3) learn a hierarchical Bayesian model that generates these embeddings in a top-down manner, from *dataset-level* concepts through *image-level* concepts to *patch-level* embeddings, and (4) infer these multi-level concepts as our multi-level conceptual explanations; to enhance ***stability***, our hierarchical Bayesian model ensures that the inferred concepts from two different perturbed versions of the same image remain similar to each other. Our contributions are:

1. We comprehensively study a systematic set of five desiderata *faithfulness, stability, sparsity, multi-level structure*, and *parsimony* when generating trustworthy concept-level explanations for ViTs.
2. We develop the first general method, dubbed ProbAbilistic Concept Explainers (PACE), as a variational Bayesian framework that satisfies these desiderata.
3. Through both quantitative and qualitative evaluations, our method demonstrates superior performance in explaining post-hoc ViT predictions via visual concepts, outperforming state-of-the-art methods across various synthetic and real-world datasets.

2. Related Work

Vision Transformers. Vision Transformers (ViTs) have revolutionized computer vision by adapting the Transformer architecture for image recognition. The pioneering ViT model processes images as sequences of patches, surpassing traditional convolutional networks in efficiency and performance (Dosovitskiy et al., 2020). Subsequent innovations include the Swin Transformer (Liu et al., 2021), which introduces a hierarchical structure with shifted windows, and the Data-efficient Image Transformers (DeiT) (Touvron et al., 2021), which optimize training with a distillation token and teacher-student strategy. The CLIP model (Radford et al., 2021) extends ViT's applicability by learning from natural language supervision, showcasing the architecture's versatility and robustness in visual representation learning.

Visual Explanation Methods. The landscape of visual explanation methods (Gilpin et al., 2018; Langer et al., 2021; Schwalbe & Finzel, 2023) in computer vision is diverse, encompassing both feature attribution and concept-based approaches. Prominent methods such as LIME and SHAP (Ribeiro et al., 2016; Lundberg & Lee, 2017; Simonyan et al., 2013; Li et al., 2021b; Shrikumar et al., 2017) provide insights by assigning importance scores to input features, enhancing the understanding of model decisions. Alongside these, concept-based explanations are also gaining popularity. *Inherent* methods (Chen et al., 2019; Alvarez Melis & Jaakkola, 2018; Koh et al., 2020; Kim et al., 2018; Chattopadhyay et al., 2024; Xu et al., 2024) learn and deduce concepts alongside the prediction model. These methods necessitate modifications to the models for explanations, posing challenges in scalability to new model architectures and increased computational demands.

To address these challenges, *post-hoc* methods (Yuksekgonul et al., 2022; Pan et al., 2021; Fel et al., 2023b; Sundararajan et al., 2017; Bach et al., 2015; Kindermans et al., 2016; Rohekar et al., 2024; Xie et al., 2022; Covert et al., 2022; Bennetot et al., 2022) deduce concepts from the existing prediction model without additional modifications. Given such advantages, our work focuses on the post-hoc setting. These methods are pivotal in image-level explanation for ViTs, providing deeper insights into ViTs' visual data processing. Nevertheless, their focus remains on image-level explanations, overlooking the multi-level structure within ViTs. They also fall short in other desiderata such as faithfulness/stability. Some methods require additional text supervision or human-annotated labels, such as (Yang et al., 2023; Ben Melech Stan et al., 2024; Menon & Vondrick, 2022; Chefer et al., 2021b;a; Ming et al., 2022; Kim et al., 2023; Losch et al., 2019). Therefore, these approaches are not applicable to our unsupervised learning setting.

In contrast, our PACE provides multi-level conceptual explanations that are faithful, stable, sparse, and parsimonious; this is verified by our empirical results in Sec. 4.

3. Methodology

In this section, we formalize the definition of five desiderata for post-hoc conceptual explanations of ViTs and describe our PACE for achieving these desiderata.

3.1. Problem Setting and Notations

Consider a dataset comprising M images, each dissected into J patches as per the model in (Dosovitskiy et al., 2020). We analyze a vision transformer, denoted as $f(\cdot)$, which processes image m (represented by $\mathbf{I}_m$) and yields: (1) the predicted label $\widehat{\mathbf{y}}_m$ with N classes, (2) patch embeddings $\mathbf{e}_m \triangleq [\mathbf{e}_{mj}]_{j=1}^J$ with $\mathbf{e}_{mj} \in \mathbb{R}^d$ (d is the hidden dimension), and (3) the attention weights $\mathbf{a}_m^{(h)} \triangleq [a_{mj}^{(h)}]_{j=1}^J$ ($\mathbf{a}_m^{(h)} \in \mathbb{R}^J$) for each patch relative to the final layer's 'CLS' token, where h signifies the attention head h. We define the mean attention weight across H heads as $a_{mj} = \frac{1}{H}\sum_{h=1}^H a_{mj}^{(h)}$, and consequently $\mathbf{a}_m \triangleq [a_{mj}]_{j=1}^J$ (refer to the ViT shown at the bottom of Fig. 1). A typical post-hoc explainer, denoted as $g(\cdot)$, processes the contextual representation $\mathbf{e}_m$, predicted label $\widehat{\mathbf{y}}_m$, and optionally ViT parameters $\mathbf{P}$, producing a

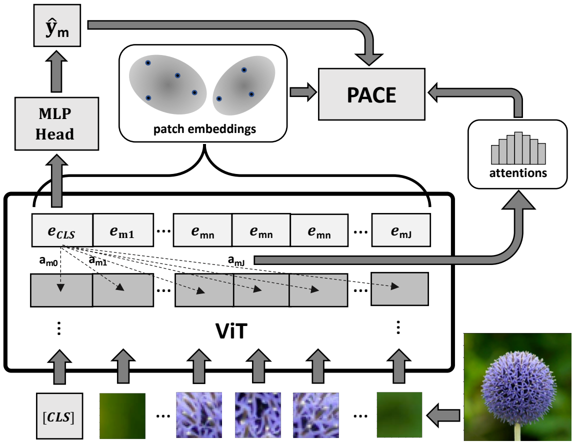

Figure 1. Overview of PACE framework. PACE utilizes patch embeddings $\mathbf{e}_m$, model predictions $\hat{\mathbf{y}}_m$, and multi-head attentions $\mathbf{a}_m$ as observations to infer hidden parameters.

concept activation $\boldsymbol{\theta}_m \in \mathbb{R}^K$ (K is the number of concepts), that is, $g(\mathbf{e}_m, \hat{\mathbf{y}}_m, \mathbf{P}) = \boldsymbol{\theta}_m$. See Appendix D for details. Note that while some methods do not inherently provide explanations with authentic concepts, the explanation activation $\boldsymbol{\theta}_m$ (or its suitably adapted version) can still be interpreted as a quasi-concept vector.

In contrast to typical post-hoc explainers that only provide image-level explanations $\boldsymbol{\theta}_m$, our PACE provides multi-level conceptual explanations; for an image m, PACE provides K *dataset-level* variables $\{\boldsymbol{\Omega}_k\}_{k=1}^K = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ ($\boldsymbol{\mu}_k \in \mathbb{R}^d$ is the Gaussian mean, and $\boldsymbol{\Sigma}_k \in \mathbb{R}^{d\times d}$ the Covariance), an *image-level* variable $\boldsymbol{\theta}_m$, and J *patch-level* variables $\boldsymbol{\phi}_m \triangleq \{\boldsymbol{\phi}_{mj}\}_{j=1}^J$ (see details in Sec. 3.2 and Sec. 4).

3.2. Definition of Trustworthy Conceptual Explanations

We formally define the five desiderata for trustworthy conceptual explanations for ViTs as follows (see Sec. 3.1).

Definition 3.1 (**Trustworthy Conceptual Explanations**). Consider a dataset $\mathcal{D}$ with M images $\mathbf{I}_m$ ($m \in 1, \ldots, M$), each consisting of J patches. For a given number of concepts K, a trustworthy conceptual explanation for an image m consists of K *dataset-level* variables $\{\boldsymbol{\Omega}_k\}_{k=1}^K = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$, an *image-level* variable $\boldsymbol{\theta}_m$, and J *patch-level* variables $\{\boldsymbol{\phi}_{mj}\}_{j=1}^J$ with the following properties:

(1) **Faithfulness**, which implies a strong relation between the concept activation $\boldsymbol{\theta}_m$ and the post-hoc label $\hat{y}_m$ derived from ViT predictions. In this paper, we measure linear faithfulness score by applying a logistic regression model $LR(\cdot)$, i.e., $\hat{y}_m = LR(\boldsymbol{\theta}_m)(1 \le m \le M)$, and evaluating its accuracy (details in Sec. 4).

(2) **Stability**, which is the consistency of explanations across different perturbed versions of the same image. For an image $\mathbf{I}_m$ with the inferred $\boldsymbol{\theta}_m$ and its perturbed version $\mathbf{I}'_m$ with inferred $\boldsymbol{\theta}'_m$, stability is quantified by $\frac{\|\boldsymbol{\theta}_m - \boldsymbol{\theta}'_m\|}{\|\boldsymbol{\theta}_m\|}$, where $\|\cdot\|$ denotes the L_2 norm.

(3) **Sparsity**, which involves the concept vector having a sparse representation, measured by the fraction of values nearing zero. Specifically, sparsity is defined as the proportion of $\boldsymbol{\theta}_m$'s entries nearing zero, i.e., $\frac{1}{K}\sum_{k=1}^K \mathbb{1}(|\theta_{mk}| < \epsilon)$, with a small threshold $\epsilon > 0$.

(4) **Multi-Level Structure**, which means that an ideal explainer should yield K *dataset-level* variables $\{\boldsymbol{\Omega}_k\}_{k=1}^K = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ representing the mean and covariance of each concept in the dataset, an *image-level* variable $\boldsymbol{\theta}_m \in \mathbb{R}^K$ for each image m, and a *patch-level* variable $\boldsymbol{\phi}_{mj} \in \mathbb{R}^K$ for each patch j in image m.

(5) **Parsimony**, which involves using the minimal number of concepts K to produce clear and simple explanations for humans. Methods with flexible concept counts could use fewer concepts while maintaining other properties' performance. In contrast, too many concepts usually lead to redundancy in conceptual explanations and a lack of compact representation.

In Definition 3.1, Property (1) ensures the learned concepts convey essential information for predicting image labels from hidden embeddings. Property (2) guarantees robustness and generalization in face of perturbations. Property (3) reflects that each prediction usually only involves a small number of relevant concepts. Property (4) offers diverse and comprehensive multi-level conceptual explanations. Finally, Property (5) facilitates learning concepts efficiently, restricts the number of redundant concepts for meaningful explanations, and reduces humans' cognitive load reading explanations. Theorem 3.2 in Sec. 3.7 provides the theoretical guarantees for PACE in terms of these properties.

3.3. ProbAbilistic Conceptual Explainers (PACE)

Drawing inspiration from hierarchical Bayesian deep learning (Wang & Yeung, 2016; 2020; Wang et al., 2016; Wang & Yan, 2023; Xu et al., 2023), we introduce a variational Bayesian framework, dubbed ProbAbilistic Concept Explainers (PACE), for post-hoc conceptual explanation of Vision Transformers (ViTs). To ensure PACE produces trustworthy concepts as defined in Definition 3.1, PACE treats the explained ViT's patch-level embeddings as observed variable and design a hierarchical Bayesian model that generates these embeddings in a top-down manner, from dataset-level concepts through image-level concepts to patch-level embeddings.

Fig. 1 shows an overview of our PACE, where patch embeddings $\mathbf{e}_m$ and ViT's predicted label $\hat{\mathbf{y}}_m$ are treated as observable variables. Attention weights $\mathbf{a}_m$ are considered as the virtual count for each patch; for example, $a_{mj} = 0.2$ means patch j is considered as $0.2J$ patches, where J is the total number of patches in image m (see details below).PACE models the patch embedding distribution using a mixture

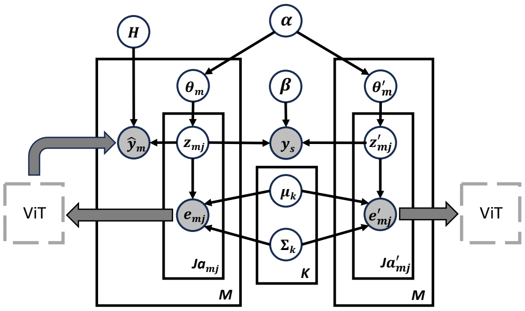

Figure 2. Graphical model of our PACE. We sample each original patch embedding $\mathbf{e}_{mj}$ for $J \cdot a_{mj}$ times, and each perturbed patch embedding $\mathbf{e}'_{mj}$ for $J \cdot a'_{mj}$ times (ViT is shared for both).

of K Gaussians (K concepts), characterized by parameters $\boldsymbol{\mu}_k \in \mathbb{R}^d$ and $\boldsymbol{\Sigma}_k \in \mathbb{R}^{d \times d}$ $(1 \leq k \leq K)$. For image m, PACE provides three levels of conceptual explanations: (1) K *dataset-level* variables $\{\boldsymbol{\Omega}_k\}_{k=1}^K = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ representing the mean and covariance of each concept k in the dataset, (2) an *image-level* variable $\boldsymbol{\theta}_m \in \mathbb{R}^K$ for each image m, and (3) J *patch-level* variable $\boldsymbol{\phi}_m \triangleq \{\boldsymbol{\phi}_{mj}\}_{j=1}^J$ for each patch j in image m, where $\boldsymbol{\phi}_{mj} \in \mathbb{R}^K$.

Generative Process. Below we describe the generative process of PACE (Fig. 2 shows the corresponding PGM):

- Draw the image-level concept distribution vector $\boldsymbol{\theta}_m \sim$ Dirichlet($\boldsymbol{\alpha}$) for either the original image $\mathbf{I}_m$ or the perturbed image $\mathbf{I}'_m$.
- For each patch j in either $\mathbf{I}_m$ or $\mathbf{I}'_m$ $(1 \leq j \leq J)$:
  - Draw the patch-level one-hot concept index $\mathbf{z}_{mj} \sim$ Categorical($\boldsymbol{\theta}_m$).
  - Given the ViT attentions a_{mj}, for $J \cdot a_{mj}$ times,
    * Draw patch j's embedding, i.e., $\mathbf{e}_{mj}$, from concept $\mathbf{z}_{mj}$'s Gaussian component $\mathbf{e}_{mj} \sim \mathcal{N}(\boldsymbol{\mu}_{z_{mj}}, \boldsymbol{\Sigma}_{z_{mj}})$.
- Draw the predicted label $\widehat{\mathbf{y}}_m \sim$ GLM($\bar{\mathbf{z}}_m, \mathbf{H}$).
- For each pair of images $\mathbf{I}_m$ and $\mathbf{I}'_m$, draw a binary variable $y_s \sim p(y_s|\bar{\mathbf{z}}_m, \bar{\mathbf{z}}'_m, \boldsymbol{\beta})$, which indicates whether $\mathbf{I}_m$ and $\mathbf{I}'_m$ come from the same image.

Here $\boldsymbol{\alpha} \in \mathbb{R}^K$ is the parameter for the Dirichlet distribution Dirichlet($\cdot$), and we define

$$\bar{\mathbf{z}}_m = {}^1\!/\!_J \sum_{j=1}^J \mathbf{z}_{mj}. \qquad (1)$$

GLM($\cdot$) denotes a categorical distribution from a generalized linear model (GLM), given by

$$p(\widehat{\mathbf{y}}_m|\mathbf{H}, \bar{\mathbf{z}}_m) = \prod_{n=1}^N [\tfrac{\exp(\boldsymbol{\eta}_n^T \bar{\mathbf{z}}_m)}{\sum_{n'} \exp(\boldsymbol{\eta}_{n'}^T \bar{\mathbf{z}}_m)}]^{\widehat{y}_{mn}}, \qquad (2)$$

where N is the number of classes, and $\mathbf{H} = [\boldsymbol{\eta}_1, ..., \boldsymbol{\eta}_N]$ are the learnable parameters ($\mathbf{H} \in \mathbb{R}^{K \times N}$). The function $p(y_s|\bar{\mathbf{z}}_m, \bar{\mathbf{z}}'_m, \boldsymbol{\beta})$ defines a distribution over whether $\mathbf{I}'_m$ is the perturbation of $\mathbf{I}_m$, where y_s is a binary label. Let $\mathcal{F} = \{1, 2, .., M\} \backslash \{m\}$, and we have $p(y_s|\bar{\mathbf{z}}_m, \bar{\mathbf{z}}'_m, \boldsymbol{\beta})$ as

$$p(y_s = 1|\bar{\mathbf{z}}_{1:M}, \bar{\mathbf{z}}'_m, \boldsymbol{\beta}) = \frac{\exp(\boldsymbol{\beta}^T (\bar{\mathbf{z}}_m \circ \bar{\mathbf{z}}'_m))}{\sum_{f \in \mathcal{F}} \exp(\boldsymbol{\beta}^T (\bar{\mathbf{z}}_m \circ \bar{\mathbf{z}}_f))}, \qquad (3)$$

where $\boldsymbol{\beta} \in \mathbb{R}^K$ is a learnable parameter, and $\circ$ denotes the element-wise product.

Given this generative process, learning the latent concept structures in ViT *across the dataset* involves learning the dataset-level parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ for the K concepts. Similarly, explaining ViT *for each image* is equivalent to inferring the distributions of the image-level and patch-level latent variables $\boldsymbol{\theta}_m$ and $\{\mathbf{z}_{mj}\}_{j=1}^J$, respectively.

3.4. Inferring Conceptual Explanations using PACE

We begin by detailing the inference of image-level and patch-level explanations (i.e., $\boldsymbol{\theta}_m$ and $\{\mathbf{z}_{mj}\}_{j=1}^J$) given the dataset-level concept parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$. We then discuss learning $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$ later in Sec. 3.5.

Inferring Patch-Level and Image-Level Concepts. Given the dataset-level concept parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$, the patch embeddings $\mathbf{e}_m \triangleq [\mathbf{e}_{mj}]_{j=1}^J$, and the associated attention weights $\mathbf{a}_m \triangleq [a_{mj}]_{j=1}^J$, as well as the predicted label $\widehat{\mathbf{y}}_m$ produced by the ViT, for each image $\mathbf{I}_m$, PACE infers the posterior distribution of the image-level concept explanation $\boldsymbol{\theta}_m$, i.e., $p(\boldsymbol{\theta}_m|\mathbf{e}_m, \mathbf{a}_m, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K, \widehat{\mathbf{y}}_m)$, and the posterior distribution of the patch-level concept explanation $\mathbf{z}_{mj}$, i.e., $p(\mathbf{z}_{mj}|\mathbf{e}_m, \mathbf{a}_m, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K, \widehat{\mathbf{y}}_m)$. Fig. 1 describes the inference process of PACE.

Variational Distributions. The aforementioned posterior distributions are intractable; hence, we employ variational inference (Jordan et al., 1998; Blei et al., 2003; Chang & Blei, 2009), using variational distributions $q(\boldsymbol{\theta}_m|\boldsymbol{\gamma}_m)$ and $q(\mathbf{z}_{mj}|\boldsymbol{\phi}_{mj})$ to approximate them. This results in the following joint variational distribution:

$$\begin{aligned} & q(\boldsymbol{\theta}_m, \{\mathbf{z}_{mj}\}_{j=1}^J|\boldsymbol{\gamma}_m, \{\boldsymbol{\phi}_{mj}\}_{j=1}^J) \\ = \ & q(\boldsymbol{\theta}_m|\boldsymbol{\gamma}_m) \cdot \prod_{j=1}^J q(\mathbf{z}_{mj}|\boldsymbol{\phi}_{mj}), \end{aligned} \qquad (4)$$

where the variational parameters $\boldsymbol{\gamma}_m \in \mathbb{R}^K$ and $\boldsymbol{\phi}_{mj} \in \mathbb{R}^K$ are estimated by maximizing Eq. 5 (more details below).

Objective Function. In line with the generative process outlined in Sec. 3.3, for each image m sampled from the dataset, the optimal variational distributions are found by maximizing the following evidence lower bound (ELBO):

$$\begin{aligned} & L(\mathbf{e}_{mj}, \boldsymbol{\gamma}_m, \boldsymbol{\phi}_m, \boldsymbol{\phi}'_m, \widehat{\mathbf{y}}_m, y_s; \boldsymbol{\alpha}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K, \mathbf{H}, \boldsymbol{\beta}) \\ & \quad = L_e + L_f + L_s, \end{aligned} \qquad (5)$$

$$\begin{aligned} L_e &= L(\mathbf{e}_{mj}, \boldsymbol{\gamma}_m, \boldsymbol{\phi}_m; \boldsymbol{\alpha}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K), \\ L_f &= L(\widehat{\mathbf{y}}_m, \boldsymbol{\phi}_m; \mathbf{H}), \\ L_s &= L(y_s, \boldsymbol{\phi}_m, \boldsymbol{\phi}'_m; \boldsymbol{\beta}). \end{aligned} \qquad (6)$$

This equation can be derived using log likelihood factorization of the variables in Fig. 2 (details provided in Appendix F.1). Below we describe each term's intuition:

1. $L_e = L(\mathbf{e}_{mj}, \boldsymbol{\gamma}_m, \boldsymbol{\phi}_m; \boldsymbol{\alpha}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K)$ is the expected log likelihood of the joint distribution of patch embeddings $\mathbf{e}_{mj}$ and the variational parameters $\boldsymbol{\gamma}_m, \boldsymbol{\phi}_m$. This term models the generation of patch embeddings $\mathbf{e}_{mj}$ in the ViT.
2. $L_f = L(\widehat{\mathbf{y}}_m, \boldsymbol{\phi}_m; \mathbf{H})$ is the expected log likelihood of the predicted label $\widehat{\mathbf{y}}_m$ given explanation $\boldsymbol{\phi}_m$. This term reflects the ***faithfulness*** property in Definition 3.1.
3. $L_s = L(y_s, \boldsymbol{\phi}_m, \boldsymbol{\phi}'_m; \boldsymbol{\beta})$ is the expected log likelihood of the binary label y_s, which indicates whether image $\mathbf{I}_m$ (with its inferred concepts $\boldsymbol{\phi}_m$) and $\mathbf{I}'_m$ (with its inferred concepts $\boldsymbol{\phi}'_m$) comes from the same image. This term reflects the ***stability*** property in Definition 3.1.

Computing L_e. We compute L_e as:

$$\begin{aligned} L_e = \ & \mathbb{E}_q[\log p(\mathbf{e}_{mj}, \boldsymbol{\gamma}_m, \boldsymbol{\phi}_m | \boldsymbol{\alpha}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K)] \\ = \ & \sum\nolimits_k \phi_{mjk} a_{mj} \log \mathcal{N}(\mathbf{e}_{mj} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) + \mathbb{E}_q[\log p(\boldsymbol{\gamma}_m, \boldsymbol{\phi}_m | \boldsymbol{\alpha})] \\ = \ & \sum\nolimits_k \phi_{mjk} a_{mj} \{-\tfrac{1}{2}(\mathbf{e}_{mj} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{e}_{mj} - \boldsymbol{\mu}_k) \\ & - \log[(2\pi)^{d/2} |\boldsymbol{\Sigma}_k|^{1/2}]\} + \mathbb{E}_q[\log p(\boldsymbol{\gamma}_m, \boldsymbol{\phi}_m | \boldsymbol{\alpha})], \end{aligned} \tag{7}$$

where the expectation is over the joint variational distribution in Eq. 4. d is the dimension of the embedding $\mathbf{e}_{mj}$.

Computing L_f. We compute L_f according to Eq. 2:

$$\begin{aligned} L_f = & \mathbb{E}_q[\log p(\widehat{\mathbf{y}}_m | \bar{\mathbf{z}}_m, \mathbf{H})] \\ = & \sum\nolimits_{n=1}^N \widehat{y}_{mn} (\boldsymbol{\eta}_n^T \bar{\boldsymbol{\phi}}_m) - \mathbb{E}_q[\log(\sum\nolimits_{n=1}^N \exp(\boldsymbol{\eta}_n^T \bar{\mathbf{z}}_m))] \\ \approx & \sum\nolimits_{n=1}^N \widehat{y}_{mn} (\boldsymbol{\eta}_n^T \bar{\boldsymbol{\phi}}_m) - \log(\sum\nolimits_{n=1}^N \exp(\boldsymbol{\eta}_n^T \bar{\boldsymbol{\phi}}_m)), \end{aligned} \tag{8}$$

where N is the number of classes for classification, n the class index. We approximate $\bar{\mathbf{z}}_m$ by taking the average of $\boldsymbol{\phi}_m$: $\bar{\mathbf{z}}_m \approx \bar{\boldsymbol{\phi}}_m = {}^1/_J \sum_{j=1}^J \boldsymbol{\phi}_{mj}$. See Appendix F.1 for details on how to approximate $\bar{\mathbf{z}}_m$. Eq. 8 implies that maximizing the log likelihood of the predicted class $\widehat{\mathbf{y}}_m$ encourages a correlation between $\widehat{\mathbf{y}}_m$ and the inferred patch-level concepts $\boldsymbol{\phi}_m$, thereby enhancing PACE's *faithfulness*.

Computing L_s. Inspired by contrastive learning (Chen et al., 2020), for each original image $\mathbf{I}_m$, we first generate its perturbed image $\mathbf{I}'_m$. Then, with their associated patch-level concepts $\bar{\mathbf{z}}_{1:M}$ and $\bar{\mathbf{z}}'_m$ from Eq. 1, the *stability* term L_s is defined as the expected likelihood of the binary label y_s in Eq. 3. Let $\mathcal{F} = \{1, .., M\} \backslash \{m\}$. We compute L_s as:

$$\begin{aligned} L_s = \ & \mathbb{E}_q[\log p(y_s = 1 | \bar{\mathbf{z}}_{1:M}, \bar{\mathbf{z}}'_m, \boldsymbol{\beta})] \\ = \ & \boldsymbol{\beta}^T (\bar{\mathbf{z}}_m \circ \bar{\mathbf{z}}'_m) - \mathbb{E}_q[\log(\sum\nolimits_{f \in \mathcal{F}} \exp(\boldsymbol{\beta}^T (\bar{\mathbf{z}}_m \circ \bar{\mathbf{z}}_f)))] \\ \approx \ & \boldsymbol{\beta}^T (\bar{\boldsymbol{\phi}}_m \circ \bar{\boldsymbol{\phi}}'_m) - \log(\sum\nolimits_{f \in \mathcal{F}} \exp(\boldsymbol{\beta}^T (\bar{\boldsymbol{\phi}}_m \circ \bar{\boldsymbol{\phi}}_f))), \end{aligned} \tag{9}$$

where $\circ$ is the element-wise product. Eq. 9 indicates that maximizing the log likelihood of y_s encourages the inferred patch-level concepts from the original and perturbed patches ($\boldsymbol{\phi}_m$ from $\mathbf{I}_m$ and $\boldsymbol{\phi}'_m$ from $\mathbf{I}'_m$) to be similar, thus enhancing PACE's *stability* against perturbations. See detailed derivations of L_e, L_f, and L_s in Appendix F.1.

Algorithm 1 Learning and Inference of PACE

```
Input: Initialized $\boldsymbol{\alpha}, \boldsymbol{\beta}, \mathbf{H}, \{\boldsymbol{\gamma}_m\}_{m=1}^M$, $\{\boldsymbol{\phi}_m\}_{m=1}^M$, $\{\boldsymbol{\Omega}_k\}_{k=1}^K$, images $\{\mathbf{I}_m\}_{m=1}^M$, perturbed images $\{\mathbf{I}'_m\}_{m=1}^M$, predicted labels $\{\widehat{\mathbf{y}}_m\}_{m=1}^M$, and number of epochs T.
for $t = 1 : T$ do
  for $m = 1 : M$ do
    Update $\boldsymbol{\phi}_m$ and $\boldsymbol{\gamma}_m$ using Eq. 10 and Eq. 11, respectively.
  Update $\{\boldsymbol{\Omega}_k\}_{k=1}^K$ using Eq. 12 and Eq. 13.
Output: $\{\boldsymbol{\Omega}_k\}_{k=1}^K$ as dataset-level, $q(\boldsymbol{\theta}_m | \boldsymbol{\gamma}_m)$ as image-level, and $q(\mathbf{z}_m | \boldsymbol{\phi}_m)$ as patch-level concept explanations.
```

Update Rules for ϕ_{mjk} and γ_{mk}. Inferring the conceptual explanations using PACE involves learning the variational parameters, ϕ_{mjk} and γ_{mk}, in Eq. 4. This is done by iteratively updating ϕ_{mjk} and γ_{mk} to maximize Eq. 5.

Specifically, taking the derivative of the ELBO in Eq. 5 with respect to ϕ_{mjk} (see Appendix F.2.1 for details) and setting it to zero, we obtain the update rule for ϕ_{mjk}:

$$\begin{aligned} \phi_{mjk} \propto \ & \tfrac{1}{|\boldsymbol{\Sigma}_k|^{1/2}} \exp[\Psi(\gamma_{mk}) - \Psi(\sum\nolimits_{k'=1}^K \gamma_{k'}) \\ & -\tfrac{1}{2} a_{mj} (\mathbf{e}_{mj} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{e}_{mj} - \boldsymbol{\mu}_k) \\ & +\tfrac{1}{J}[(\sum\nolimits_{n=1}^N \widehat{y}_{ml} \boldsymbol{\eta}_l) - \tfrac{\sum_{n'=1}^N \exp(\boldsymbol{\eta}_{n'}^T \bar{\boldsymbol{\phi}}_m) \boldsymbol{\eta}_n}{\sum_{n=1}^N \exp(\boldsymbol{\eta}_n^T \bar{\boldsymbol{\phi}}_m)} \\ & +\boldsymbol{\beta}^T \bar{\boldsymbol{\phi}}'_m - \tfrac{\sum_{f \in \mathcal{F}} \exp(\boldsymbol{\beta}^T (\bar{\boldsymbol{\phi}}_m \circ \bar{\boldsymbol{\phi}}_f))(\boldsymbol{\beta}^T \bar{\boldsymbol{\phi}}_f)}{\sum_{f \in \mathcal{F}} \exp(\boldsymbol{\beta}^T (\bar{\boldsymbol{\phi}}_m \circ \bar{\boldsymbol{\phi}}_f))})], \end{aligned} \tag{10}$$

with the normalization constraint $\sum_{k=1}^K \phi_{mjk} = 1$. Here $\Psi(\cdot)$ is the digamma function (the first derivative of the logarithm of the Gamma function $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$).

Similarly, the update rule for γ_{mk} is:

$$\gamma_{mk} = \alpha_k + \sum\nolimits_{j=1}^J \phi_{mjk} a_{mj}. \tag{11}$$

In summary, the inference algorithm alternates between updating ϕ_{mjk} for all (m, j, k) tuples and updating $\boldsymbol{\gamma}_{mk}$ for all (m, k) tuples until convergence.

Image- and Patch-Level Explanations: $\boldsymbol{\theta}_m$ and ϕ_{mj}. We then use $\boldsymbol{\gamma}_m = \{\gamma_{mk}\}_{k=1}^K$ with $q(\boldsymbol{\theta}_m | \boldsymbol{\gamma}_m)$ to obtain the *image-level* conceptual explanation $\boldsymbol{\theta}_m$ and use $\boldsymbol{\phi}_{mj} = \{\phi_{mjk}\}_{k=1}^K$ as the *patch-level* explanation.

3.5. Learning of PACE

Learning Dataset-Level Explanations: $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$. The inference algorithm in Sec. 3.4 assumes the availability of the dataset-level concept parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$.

To learn these parameters, one needs to iterate between (1) inferring image-level and patch-level variational parameters $\boldsymbol{\gamma}_m$ and $\boldsymbol{\phi}_{mj}$ in Sec. 3.4, respectively, and (2) learning dataset-level concept parameters $\{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$. Alg. 1 summarizes the learning of PACE.

Update Rules for $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$. Similar to Sec. 3.4, we expand the ELBO in Eq. 7 (see Appendix F.2.2 for details) and set its derivative with respect to $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ to zero, yielding the update rule for learning $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$:

$$\boldsymbol{\mu}_k = \frac{\sum_{m,j} \phi_{mjk} a_{mj} \mathbf{e}_{mj}}{\sum_{m,j} \phi_{mjk} a_{mj}}, \tag{12}$$

$$\boldsymbol{\Sigma}_k = \frac{\sum_{m,j} \phi_{mjk} a_{mj} (\mathbf{e}_{mj} - \boldsymbol{\mu}_k)(\mathbf{e}_{mj} - \boldsymbol{\mu}_k)^T}{\sum_{m,j} \phi_{mjk} a_{mj}}. \tag{13}$$

3.6. Summary of Learning and Inference of PACE

In summary, PACE is a variational Bayesian framework that consists of (1) the **learning stage** to train on the training set, and (2) the **inference stage** to explain on the test set.

For example, given a finetuned ViT classifier on a dataset, PACE explains it by (1) training the global parameters, i.e., the *dataset-level* concept centers $\boldsymbol{\mu}_k$ and covariance matrices $\boldsymbol{\Sigma}_k$ (where $k = 1, \dots, K$) as dataset-level explanations, on the training set (these are called *global* parameters because they are shared across all data points, e.g. images), and (2) inferring the *local* parameters, i.e., the *image-level* concepts (explanations) $\boldsymbol{\theta}_m$ and *patch-level* concepts (explanations) $\boldsymbol{\phi}_m$, on the test set (these are called *local* parameters because each image has its own $\boldsymbol{\theta}_m$ and $\boldsymbol{\phi}_m$).

Below, we discuss the learning and inference processes, respectively.

The Learning Stage. In Sec. 3.5, we describe the process of learning the *global parameters* $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ (where $k = 1, \dots, K$). As described in Alg. 1, in each epoch t ($t = 1, \dots, T$):

(1) PACE first infers the *local* parameters $\boldsymbol{\gamma}_m$ and $\boldsymbol{\phi}_{mj}$ for each document m using Eq. 10 and Eq. 11;

(2) PACE then updates the *global* parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ (where $k = 1, \dots, K$) for the entire dataset using Eq. 12 and Eq. 13.

The learning stage concludes at the T^{th} epoch. Note that the process above is iterative; it alternates between (1) updating the *local* parameters and (2) updating the *global* parameters.

The Inference Stage. In Sec. 3.4, we describe the process of inferring the *local* parameters $\boldsymbol{\theta}_m$ and $\boldsymbol{\phi}_{mj}$ after the PACE learns the global parameters in the learning stage. Specifically, given the *global* parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ (where $k = 1, \dots, K$),

(1) PACE initializes *local* parameters $\boldsymbol{\gamma}_m$ and $\boldsymbol{\phi}_{mj}$;

(2) Given the current $\boldsymbol{\gamma}_m$ and $\boldsymbol{\phi}_{mj}$, PACE updates the *local parameters* $\boldsymbol{\gamma}_m$ using Eq. 10;

(3) Given the current $\boldsymbol{\gamma}_m$ and $\boldsymbol{\phi}_{mj}$, PACE updates the *local parameters* $\boldsymbol{\phi}_{mj}$ using Eq. 11;

(4) PACE repeats (2) and (3) until $\boldsymbol{\gamma}_m$ and $\boldsymbol{\phi}_{mj}$ converge;

(5) PACE then infers the image-level concept $\boldsymbol{\theta}_m$ using the learned variational distribution $q(\boldsymbol{\theta}_m|\boldsymbol{\gamma}_m)$, which is a Dirichlet distribution. One can (roughly) think of $\boldsymbol{\theta}_m$ as a normalized version of $\boldsymbol{\gamma}_m$.

3.7. Discussion and Theoretical Analysis

Our PACE addresses all five desiderata in Definition 3.1:

- **Faithfulness** is encouraged by maximizing the prediction $\widehat{\mathbf{y}}_m$'s likelihood, i.e., L_f in Eq. 8.
- **Stability** against perturbations is enhanced by maximizing the binary label y_s's likelihood L_s in Eq. 9.
- **Sparsity** is encouraged by the Dirichlet prior $p(\boldsymbol{\theta}_m|\boldsymbol{\alpha})$ that regularizes the inference of $\boldsymbol{\theta}_m$.
- **Multi-Level Structure** is intrinsically supported by our multi-level generative process in Sec. 3.3.
- **Parsimony** is ensured by the flexibility in choosing the number of concepts K in PACE (see Appendix A).

Theorem 3.2 below further demonstrates that PACE's inferred image-level and patch-level explanations, $\boldsymbol{\theta}_m$ and $\{\boldsymbol{\phi}_{mj}\}_{j=1}^J$, align with the properties in Definition 3.1.

Theorem 3.2 (**Mutual Information Maximization**). *The ELBO in Eq. 5 is upper-bounded by the sum of (1) mutual information between contextual embeddings $\mathbf{e}_m$ and multi-level explanation $\boldsymbol{\theta}_m, \{\boldsymbol{\phi}_{mj}\}_{j=1}^J$ in Definition 3.1, (2) mutual information between the predicted label $\widehat{\mathbf{y}}_m$ and patch-level concept $\boldsymbol{\phi}_m$, and (3) mutual information between the patch-level original concept $\boldsymbol{\phi}_m$ and perturbed concept $\boldsymbol{\phi}'_m$. Formally, with approximate posteriors $q(\boldsymbol{\theta}_m|\boldsymbol{\gamma}_m)$ and $q(\mathbf{z}_{mj}|\boldsymbol{\phi}_{mj})$, we have*

$$\begin{aligned} & L(\mathbf{e}_{mj}, \boldsymbol{\gamma}_m, \boldsymbol{\phi}_m, \boldsymbol{\phi}'_m, \widehat{\mathbf{y}}_m, y_s) \\ \leq\; & I(\mathbf{e}_m; \boldsymbol{\theta}_m, \boldsymbol{\phi}_m) + I(\widehat{\mathbf{y}}_m; \boldsymbol{\phi}_m) + I(\boldsymbol{\phi}_m; \boldsymbol{\phi}'_m) + C, \end{aligned} \tag{14}$$

where the C is a constant.

The proof of Theorem 3.2 is provided in Appendix E. Theorem 3.2 implies that maximizing the ELBO in Eq. 5 is equivalent to maximizing the sum of (1) mutual information between the contextual embeddings $\mathbf{e}_m$ and ***multi-level*** conceptual explanations defined in Definition 3.1, thereby ensuring the generated explanations are informative, (2) mutual information between the ViT prediction $\widehat{\mathbf{y}}_m$ and patch-level concept $\boldsymbol{\phi}_m$, thereby enhancing PACE's ***faithfulness***, and (3) mutual information between the patch-level original $\boldsymbol{\phi}_m$ and perturbed $\boldsymbol{\phi}'_m$, thereby enhancing PACE's ***stability*** against perturbations.

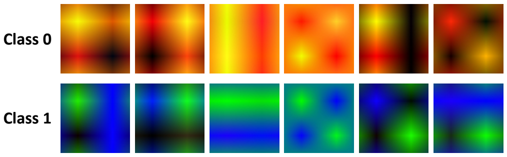

Figure 3. Example images from our *Color* dataset.

4. Experiments

In this section, we compare PACE with existing methods on one synthetic dataset and three real-world datasets.

4.1. Datasets

We constructed *Color* as a synthetic dataset with clear definition of 4 concepts (red/yellow/green/blue). It contains two image classes: *Class 0* (images with red/yellow colors) and *Class 1* (green/blue colors), both against a *black* background (see example images in Fig. 3 and more details in Appendix A). We use three real-world datasets, *Oxford 102 Flower (Flower)* (Nilsback & Zisserman, 2008), *Stanford Cars (Cars)* (Krause et al., 2013), and *CUB-200-2011 (CUB)* (Wah et al., 2011) (for dataset statistics, see Appendix A). For real-world datasets, we follow the preprocessing steps from (Dosovitskiy et al., 2020) and use the same train-test split. For the *Color* dataset, we adopt an 8:2 train/test split among 2,000 images (1,000 per class).

4.2. Baselines

We compare PACE with state-of-the-art methods, including:

- **SHAP** (Lundberg & Lee, 2017) is an explanation method that assigns importance scores to input features using Shapley values.
- **LIME** (Ribeiro et al., 2016) explains the model by approximates it with a local surrogate model via data perturbation.
- **SALIENCY** (Simonyan et al., 2013) uses the saliency map of an image to explain the model prediction.
- **AGI** (Pan et al., 2021) produces explanations via adversarial gradient integration.
- **CRAFT** (Fel et al., 2023b) employs recursive low-rank matrix factorization to obtain concepts from intermediate layers.

4.3. Evaluation Metrics

With ViT-Base (Dosovitskiy et al., 2020) as the prediction model, we evaluate different methods against the five desiderata defined in Definition 3.1:

- **(Linear) Faithfulness:** We fit a logistic regression (LR) model $\widehat{\mathbf{y}}_m = LR(\boldsymbol{\theta}_m)$ to each dataset's training set, with $\widehat{\mathbf{y}}_m$ as the prediction of the ViT, and test the model's accuracy on the test set. Higher accuracy indicates stronger (linear) faithfulness. Note that one can fit more complex models (e.g., nonlinear models such as neural networks) to evaluate nonlinear faithfulness; for simplicity, we focus on linear faithfulness. **Stability:** For each input image $\mathbf{I}_m$ in the test set, we generate an augmented version $\mathbf{I}'_m$ (following Chen et al. (2020)), and compute the normalized difference between inferred concepts $\boldsymbol{\theta}_m$ and $\boldsymbol{\theta}'_m$, i.e., $\frac{\|\boldsymbol{\theta}_m - \boldsymbol{\theta}'_m\|}{\|\boldsymbol{\theta}_m\|}$. Lower values indicate stronger stability.
- **Sparsity:** We compute sparsity (with the threshold $\epsilon = 0.1/K$) as $\frac{1}{K}\sum_{k=1}^{K} \mathbb{1}(|\theta_{mk}| < \epsilon)$, where K is the number of concepts. For many concept-based explanation methods, including ours, the inferred activation typically normalizes to sum up to 1. If this is not the case, we normalize the explanation activation before calculating the sparsity score, ensuring a fair comparison.
- **Multi-Level Structure:** As highlighted in Sec. 1, baseline models do not account for dataset-level and/or patch-level concepts, thereby possess *No* or *Partial* multi-level structure. In contrast, PACE is specifically designed to offer *Full* conceptual explanations at three levels: dataset, image, and patch. We will demonstrate in Sec. 4.5 that modeling embeddings' distribution is instrumental in bridging three levels of ViT concepts.
- **Parsimony:** For the conceptual explanation methods PACE and CRAFT (Fel et al., 2023b), we set the number of concepts $K = 100$, to facilitate a fair comparison. Note that other baseline models' number of concept K is constrained to the hidden dimension of ViT embeddings, i.e., $K = 768$; they therefore fall short in parsimony.

For details and the three other desiderata, see Appendix A.

4.4. Quantitative Results

Table 1 shows the quantitative results for our PACE and the baselines for the desiderata in Definition 3.1 across one synthetic dataset (*Color*) and three real-world datasets (*Flower*, *Cars*, and *CUB*). For a detailed discussion on Multi-level Structure and Parsimony, please refer to **Appendix A**. Below we discuss Faithfulness, Stability, and Sparsity in detail. ***Color*.** On the *Color* dataset, PACE distinctly surpasses other leading models, as detailed in Table 1. PACE achieves perfect faithfulness (1.00) and the best stability score (0.20), demonstrating consistency in its explanations. It leads in sparsity (0.97), delivering focused and clear explanations.

***Flower*, *Cars*, and *CUB*.** Our evaluation on three real-world datasets – *Flower*, *Cars*, and *CUB* – reveals PACE's significant advantages over established baselines across various desiderata. As shown in Table 1, PACE consistently registers the highest faithfulness scores (0.80 on *Flower*, 0.50 on *Cars*, and 0.56 on *CUB*), reflecting its superior precision in

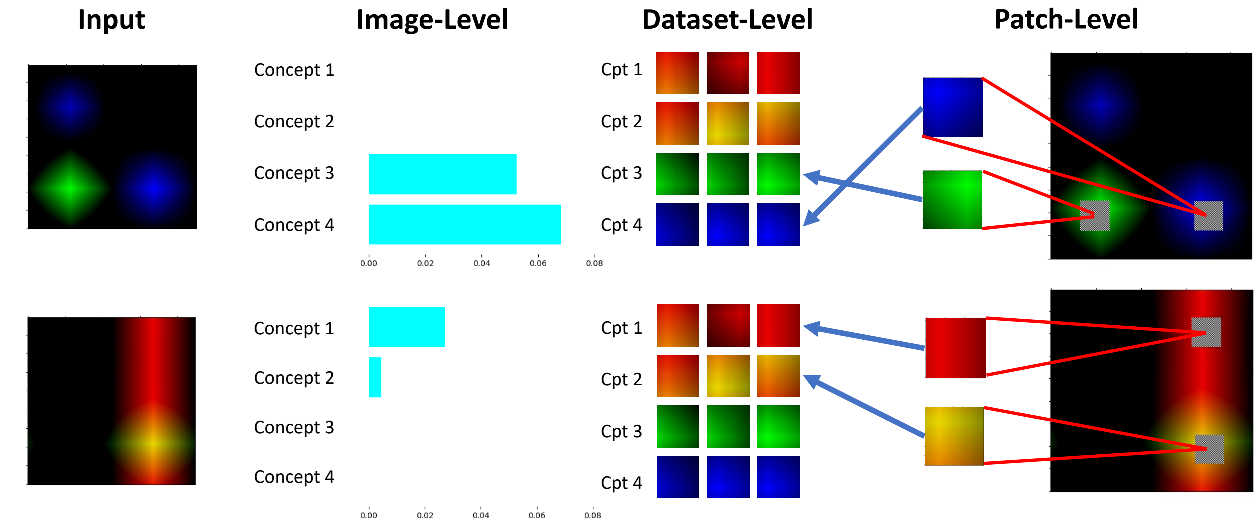

Figure 4. PACE's three-level conceptual explanations on the *Color* dataset. **Dataset-Level**: PACE's top 4 dataset-level concepts; for each concept k, we plot the top 3 patches with $\mathbf{e}_{mj}$ closest to $\boldsymbol{\mu}_k$. **Image-Level**: Given an input image m, we show PACE's generated image-level explanation θ_m for the 4 selected concepts. For example, for the top-left input image, PACE's generated image-level explanation θ_m indicates a strong association with Concept 3 (green) and Concept 4 (blue). **Patch-Level**: Given an input image m, PACE's ϕ_{mj} identifies the top concepts for the selected patches. For example, for the top-left input image, the blue patch is associated with Concept 4 (containing similar blue patches across the dataset) while the green patch is linked to Concept 3 (containing similar green patches across the dataset).

Table 1. Results for the five desiderata in Definition 3.1 for different methods on four datasets. 'All' denotes results on all four datasets. We mark the best results with **bold face** and the second best results with <u>underline</u>. ↑ / ↓ indicates higher/lower is better, respectively.

Desiderata	(Linear) Faithfulness ↑				Stability ↓				Sparsity ↑				Multi-Level	Parsimony ↓
Datasets	*Color*	*Flower*	*Cars*	*CUB*	*Color*	*Flower*	*Cars*	*CUB*	*Color*	*Flower*	*Cars*	*CUB*	All	All
SHAP	0.47	0.52	0.44	0.34	4.39	0.92	1.21	1.55	0.54	0.12	0.13	0.11	No	768
LIME	0.54	0.06	0.02	0.03	1.50	1.54	1.45	1.80	<u>0.59</u>	**0.54**	**0.52**	<u>0.55</u>	No	768
Saliency	**1.00**	<u>0.57</u>	**0.50**	<u>0.49</u>	<u>0.35</u>	<u>0.47</u>	<u>0.43</u>	<u>0.48</u>	0.01	0.00	0.00	0.00	No	768
AGI	**1.00**	0.54	0.34	<u>0.49</u>	1.40	1.21	1.83	2.53	0.01	0.07	0.04	0.03	No	768
CRAFT	0.59	0.01	0.01	0.00	4.49	0.52	1.76	0.64	0.26	0.29	0.11	0.25	Partial	**100**
PACE	**1.00**	**0.80**	**0.50**	**0.56**	**0.20**	**0.12**	**0.05**	**0.05**	**0.97**	<u>0.48</u>	<u>0.49</u>	**0.63**	**Full**	**100**

Table 2. Average results across all four datasets in terms of faithfulness, stability, and sparsity for different methods. The best results are marked with **bold face**. ↑ / ↓ indicates higher/lower is better, respectively.

Desiderata	SHAP	LIME	Saliency	AGI	CRAFT	PACE
Faithfulness ↑	0.44	0.16	0.64	0.59	0.15	**0.72**
Stability ↓	2.02	1.57	0.43	1.74	1.85	**0.11**
Sparsity ↑	0.23	0.55	0.00	0.04	0.23	**0.64**

mirroring the model's decision-making process; note that both *Cars* and *CUB* contain a large number of classes (196 and 200); therefore, a linear faithfulness score of 0.50 is already very high. As mentioned in Sec. 4.3, one can always fit more complex models (e.g., nonlinear models such as a two-layer neural networks) to evaluate nonlinear faithfulness; for simplicity, we focus on linear faithfulness in this paper. Its stability scores (lower is better) on these three datasets are 0.12, 0.05, and 0.05, respectively, illustrating its resilience to input perturbations. In terms of sparsity, PACE is highly competitive, achieving second-best results and thus providing succinct, pertinent explanations.

Average Performance Across Datasets. Table 2 shows the average performance across all four datasets in terms of faithfulness, stability, and sparsity. PACE consistently leads in faithfulness (0.72), stability (0.11), and sparsity (0.64). Compared to other models, its improvements are substantial, enhancing faithfulness by $0.08 \sim 0.57$, stability by $0.32 \sim 1.91$, and sparsity by $0.09 \sim 0.64$, verifying PACE's effectiveness in terms of trustworthy explanations.

4.5. Qualitative Analysis

***Color*.** Fig. 4 illustrates PACE's three-level explanations on the *Color* dataset, where the ViT correctly predicts the top and bottom input images as *Class 1* (green/blue colors) and *Class 0* (images with red/yellow colors), respectively.

- **Dataset-Level Explanation:** The dataset-level column shows PACE's top 4 dataset-level concepts (for each concept, we plot the top 3 patches with $\mathbf{e}_{mj}$ closest to $\boldsymbol{\mu}_k$); they are consistent with the 4 primary colors in the dataset (see Sec. 4.1), verifying the PACE's effec-

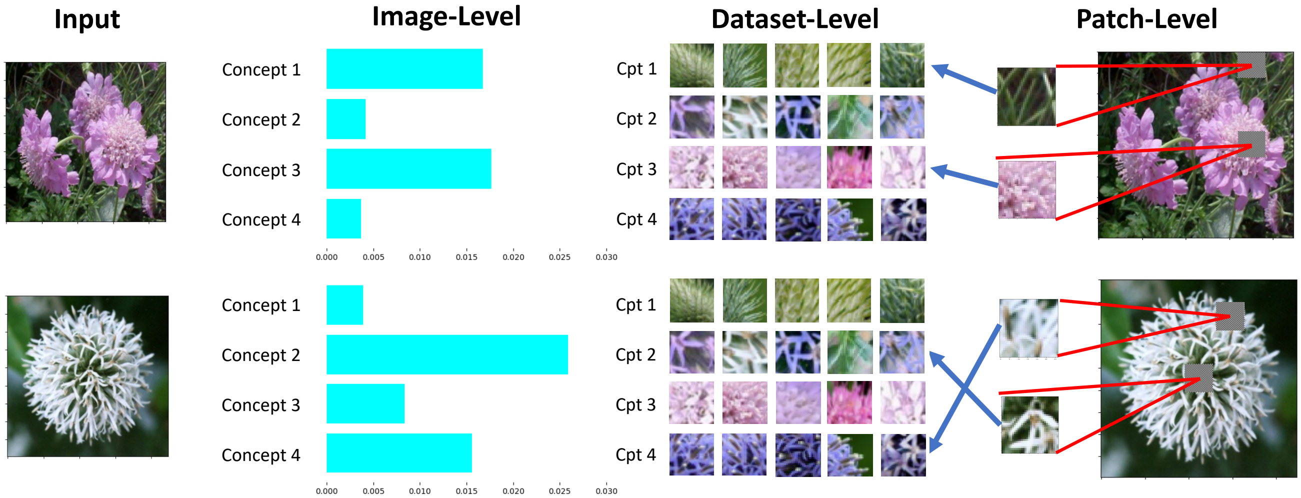

Figure 5. PACE's three-level conceptual explanations on the *Flower* dataset. **Dataset-Level**: PACE's top 4 dataset-level concepts (i.e., 'Cpt 1' to 'Cpt 4'); for each concept, we plot the top 5 patches with $\mathbf{e}_{mj}$ closest to $\boldsymbol{\mu}_k$. **Image-Level**: Given an input image m, e.g., the top-left image, PACE's generated image-level explanation $\boldsymbol{\theta}_m$ indicates a strong association with Concepts 1 (green stem/leaves) and 3 (purple petal). This is consistent with the image's petal as the foreground and stem/leaves as the background. **Patch-Level**: For an input image m, e.g., the top-left image, PACE's $\boldsymbol{\phi}_{mj}$ identifies the top concepts for patch j. The patch at the top (green stem/leaves) is associated with Concept 1 (containing similar appearance patches across the dataset); the middle patch (purple petal) is linked to Concept 3 (containing patches of other pincushion flower petal across the dataset).

tiveness.

- **Image-Level Explanation:** Given an input image m, PACE infers the image-level concepts. For example, given the input image in the top row of Fig. 4, PACE's generated image-level explanation θ_m indicates a strong association with Concept 3 (green) and Concept 4 (blue). This is consistent with the color distribution in the image m, predominantly blue and less green.
- **Patch-Level Explanation:** PACE also generated patch-level explanation. For the same image above, PACE's $\boldsymbol{\phi}_{mj}$ identifies the top concepts for the selected patches; the blue patch is associated with Concept 4 (containing similar blue patches across the dataset), while the green patch is linked to Concept 3 (containing green patches across the dataset).

***Flower*.** Fig. 5 demonstrates PACE's three-level explanations on the *Flower* dataset.

- **Dataset-Level Explanation:** The dataset-level column shows the top 4 dataset-level concepts from our PACE, each with unique shapes, texture, and colors.
- **Image-Level Explanation:** Given an input image m, PACE infers the image-level concepts. For example, given the input image in the top row of Fig. 5, PACE's generated image-level explanation $\boldsymbol{\theta}_m$ indicates a strong association with Concepts 1 (green stem/leaves) and 3 (purple petal). This is consistent with the image's petal as the foreground and stem/leaves as the background.
- **Patch-Level Explanation:** PACE also generated patch-level explanation. For the same image m above, PACE's $\boldsymbol{\phi}_{mj}$ identifies the top concepts for patch j. The patch at the top (green stem/leaves) is associated with Concept 1, comprising similar appearance patches across the dataset; the middle patch (purple petal) is linked to Concept 3, which includes patches of other pincushion flower petal across the dataset.

See Appendix C for further qualitative analysis on more real-world datasets.

5. Conclusion

In this paper, we identify five desiderata *faithfulness*, *stability*, *sparsity*, *multi-level structure*, and *parsimony* when generating trustworthy concept-level explanations for ViTs. We develop the first general method, PACE, that is compatible with any transformer variants and satisfies these desiderata. Through both quantitative and qualitative evaluations, our method demonstrates superior performance in explaining post-hoc ViT predictions via visual concepts, outperforming state-of-the-art methods across various datasets. As a limitation, our approach assumes a fixed number of concepts (Similar to existing methods). Therefore future work could focus on developing PACE into a non-parametric explainer that automatically determines the number of concepts. Another limitation is that our approach requires access to hidden states and attention weights from the layers inside ViTs; we argue that this is an advantage because it allows our PACE to interpret vision foundation models' internals thoroughly rather than simply their output superficially.

Acknowledgements

We extend our heartfelt thanks to Akshay Nambi and Tanuja Ganu from Microsoft Research for their invaluable suggestions, which greatly improved the presentation of this paper. We are grateful for the support from Microsoft Research AI & Society Fellowship, NSF Grant IIS-2127918, and Amazon Faculty Research Award. Additionally, we thank the reviewers and the area chair/senior area chair for their insightful feedback and for recognizing the novelty and contributions of our work. We thank the Center for AI Safety (CAIS) for making computing resources available for this research.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Agarwal, C., Krishna, S., Saxena, E., Pawelczyk, M., Johnson, N., Puri, I., Zitnik, M., and Lakkaraju, H. Openxai: Towards a transparent evaluation of model explanations. *Advances in Neural Information Processing Systems*, 35: 15784–15799, 2022.

Alvarez Melis, D. and Jaakkola, T. Towards robust interpretability with self-explaining neural networks. *Advances in neural information processing systems*, 31, 2018.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., and Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

Ben Melech Stan, G., Yehezkel Rohekar, R., Gurwicz, Y., Olson, M. L., Bhiwandiwalla, A., Aflalo, E., Wu, C., Duan, N., Tseng, S.-Y., and Lal, V. Lvlm-intrepret: An interpretability tool for large vision-language models. *arXiv e-prints*, pp. arXiv–2404, 2024.

Bennetot, A., Franchi, G., Del Ser, J., Chatila, R., and Diaz-Rodriguez, N. Greybox xai: A neural-symbolic learning framework to produce interpretable predictions for image classification. *Knowledge-Based Systems*, 258:109947, 2022.

Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *the Journal of machine Learning research*, 3: 993–1022, 2003.

Chang, J. and Blei, D. Relational topic models for document networks. In *Artificial intelligence and statistics*, pp. 81–88. PMLR, 2009.

Chattopadhyay, A., Chan, K. H. R., and Vidal, R. Bootstrapping variational information pursuit with large language and vision models for interpretable image classification. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=9bmTbVaA2A.

Chefer, H., Gur, S., and Wolf, L. Generic attention-model explainability for interpreting bi-modal and encoder-decoder transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 397–406, 2021a.

Chefer, H., Gur, S., and Wolf, L. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 782–791, 2021b.

Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., and Su, J. K. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

Chen, R., Li, J., Zhang, H., Sheng, C., Liu, L., and Cao, X. Sim2word: Explaining similarity with representative attribute words via counterfactual explanations. *ACM Transactions on Multimedia Computing, Communications and Applications*, 19(6):1–22, 2023.

Chen, R., Zhang, H., Liang, S., Li, J., and Cao, X. Less is more: Fewer interpretable region via submodular subset selection. *arXiv preprint arXiv:2402.09164*, 2024.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

Colin, J., Fel, T., Cadène, R., and Serre, T. What i cannot predict, i do not understand: A human-centered evaluation framework for explainability methods. *Advances in Neural Information Processing Systems*, 35:2832–2845, 2022.

Covert, I., Kim, C., and Lee, S.-I. Learning to estimate shapley values with vision transformers. *arXiv preprint arXiv:2206.05282*, 2022.

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*, 2020.

Fel, T., Boutin, V., Moayeri, M., Cadène, R., Bethune, L., Chalvidal, M., Serre, T., et al. A holistic approach to unifying automatic concept extraction and concept importance estimation. *arXiv preprint arXiv:2306.07304*, 2023a.

Fel, T., Picard, A., Bethune, L., Boissin, T., Vigouroux, D., Colin, J., Cadène, R., and Serre, T. Craft: Concept recursive activation factorization for explainability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2711–2721, 2023b.

Ghorbani, A., Wexler, J., Zou, J. Y., and Kim, B. Towards automatic concept-based explanations. *Advances in neural information processing systems*, 32, 2019.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pp. 80–89. IEEE, 2018.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. In *Learning in graphical models*, pp. 105–161. Springer, 1998.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Kim, S., Oh, J., Lee, S., Yu, S., Do, J., and Taghavi, T. Grounding counterfactual explanation of image classifiers to textual concept space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10942–10950, 2023.

Kindermans, P.-J., Schütt, K., Müller, K.-R., and Dähne, S. Investigating the influence of noise and distractors on the interpretation of neural networks. *arXiv preprint arXiv:1611.07270*, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Koh, P. W., Nguyen, T., Tang, Y. S., Mussmann, S., Pierson, E., Kim, B., and Liang, P. Concept bottleneck models. In *International conference on machine learning*, pp. 5338–5348. PMLR, 2020.

Krause, J., Stark, M., Deng, J., and Fei-Fei, L. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pp. 554–561, 2013.

Langer, M., Oster, D., Speith, T., Hermanns, H., Kästner, L., Schmidt, E., Sesing, A., and Baum, K. What do we want from explainable artificial intelligence (xai)?–a stakeholder perspective on xai and a conceptual model guiding interdisciplinary xai research. *Artificial Intelligence*, 296:103473, 2021.

Li, J., Kuang, K., Li, L., Chen, L., Zhang, S., Shao, J., and Xiao, J. Instance-wise or class-wise? a tale of neighbor shapley for concept-based explanation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 3664–3672, 2021a.

Li, J., Zhang, C., Zhou, J. T., Fu, H., Xia, S., and Hu, Q. Deep-lift: Deep label-specific feature learning for image annotation. *IEEE transactions on Cybernetics*, 52(8): 7732–7741, 2021b.

Li, X.-H., Shi, Y., Li, H., Bai, W., Song, Y., Cao, C. C., and Chen, L. Quantitative evaluations on saliency methods: An experimental study. *arXiv preprint arXiv:2012.15616*, 2020.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

Losch, M., Fritz, M., and Schiele, B. Interpretability beyond classification output: Semantic bottleneck networks. *arXiv preprint arXiv:1907.10882*, 2019.

Lundberg, S. M. and Lee, S.-I. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Menon, S. and Vondrick, C. Visual classification via description from large language models. *arXiv preprint arXiv:2210.07183*, 2022.

Ming, Y., Cai, Z., Gu, J., Sun, Y., Li, W., and Li, Y. Delving into out-of-distribution detection with vision-language representations. *Advances in neural information processing systems*, 35:35087–35102, 2022.

Murdoch, W. J., Singh, C., Kumbier, K., Abbasi-Asl, R., and Yu, B. Definitions, methods, and applications in interpretable machine learning. *Proceedings of the National Academy of Sciences*, 116(44):22071–22080, 2019.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pp. 722–729. IEEE, 2008.

Novello, P., Fel, T., and Vigouroux, D. Making sense of dependence: Efficient black-box explanations using dependence measure, september 2022. *arXiv preprint arXiv:2206.06219*.

Oikarinen, T., Das, S., Nguyen, L. M., and Weng, T.-W. Label-free concept bottleneck models. *arXiv preprint arXiv:2304.06129*, 2023.

Pan, D., Li, X., and Zhu, D. Explaining deep neural network models with adversarial gradient integration. In *Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Ribeiro, M. T., Singh, S., and Guestrin, C. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144, 2016.

Rohekar, R. Y., Gurwicz, Y., and Nisimov, S. Causal interpretation of self-attention in pre-trained transformers. *Advances in Neural Information Processing Systems*, 36, 2024.

Schwalbe, G. and Finzel, B. A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts. *Data Mining and Knowledge Discovery*, pp. 1–59, 2023.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In *International conference on machine learning*, pp. 3145–3153. PMLR, 2017.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Wah, C., Branson, S., Welinder, P., Perona, P., and Belongie, S. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

Wang, A., Lee, W.-N., and Qi, X. Hint: Hierarchical neuron concept explainer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10254–10264, 2022.

Wang, B., Li, L., Nakashima, Y., and Nagahara, H. Learning bottleneck concepts in image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10962–10971, 2023.

Wang, H. and Yan, J. Self-interpretable time series prediction with counterfactual explanations. In *ICML*, 2023.

Wang, H. and Yeung, D.-Y. Towards bayesian deep learning: A framework and some existing methods. *TDKE*, 28(12): 3395–3408, 2016.

Wang, H. and Yeung, D.-Y. A survey on bayesian deep learning. *CSUR*, 53(5):1–37, 2020.

Wang, H., Xingjian, S., and Yeung, D.-Y. Natural-parameter networks: A class of probabilistic neural networks. In *NIPS*, pp. 118–126, 2016.

Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., and Hu, X. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 24–25, 2020.

Xie, W., Li, X.-H., Cao, C. C., and Zhang, N. L. Vit-cx: Causal explanation of vision transformers. *arXiv preprint arXiv:2211.03064*, 2022.

Xu, X., Qin, Y., Mi, L., Wang, H., and Li, X. Energy-based concept bottleneck models: Unifying prediction, concept intervention, and probabilistic interpretations. In *ICLR*, 2024.

Xu, Z., Hao, G., He, H., and Wang, H. Domain indexing variational bayes: Interpretable domain index for domain adaptation. In *ICLR*, 2023.

Yang, Y., Panagopoulou, A., Zhou, S., Jin, D., Callison-Burch, C., and Yatskar, M. Language in a bottle: Language model guided concept bottlenecks for interpretable image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19187–19197, 2023.

Yuksekgonul, M., Wang, M., and Zou, J. Post-hoc concept bottleneck models. *arXiv preprint arXiv:2205.15480*, 2022.

Zhang, R., Madumal, P., Miller, T., Ehinger, K. A., and Rubinstein, B. I. Invertible concept-based explanations for cnn models with non-negative concept activation vectors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11682–11690, 2021.

A. Implementation Details

In this section, we provide implementation details of PACE.

***Color* Dataset Generation.** We constructed *Color* as a synthetic dataset with clear definition of 4 concepts (red/yellow/green/blue). It contains two image classes: *Class 0* (images with red/yellow colors) and *Class 1* (green/blue colors), both against a *black* background (see example images in Fig. 3). Images are initially created at a 2 × 2 resolution, where each pixel samples color from (red/yellow,green/blue,black), and are subsequently up-sampled with gradual color shift to 224 × 224 for ViT inputs. We introduce random Gaussian noise to each image. The dataset includes 1000 images per class, with a split of 800 training and 200 test samples.

Table 3. Dataset statistics, i.e., the number of train/test images (M_{train}/M_{test}), the number of classes N, and the number of patches J per image.

Dataset	M_{train}	M_{test}	N	J
Color	1,600	400	2	197
Flower	7,169	1,020	102	197
Cars	8,144	8,041	196	197
CUB	5,994	5,794	200	197

Table 3 provides statistics for the COLOR dataset along with three additional real-world datasets.

Experimental Setup. Following the approach outlined in (Chen et al., 2020), we implement perturbation described in Definition 3.1 based on their augmentation algorithms, as demonstrated by the following code snippet:

```
contrast_transforms = transforms.Compose([transforms.RandomHorizontalFlip(),
                                          transforms.RandomResizedCrop(size=size),
                                          transforms.RandomApply([
                                              transforms.ColorJitter(brightness=0.5,
                                                                     contrast=0.5,
                                                                     saturation=0.5,
                                                                     hue=0.1)
                                          ], p=0.8),
                                          transforms.RandomGrayscale(p=0.2),
                                          transforms.GaussianBlur(kernel_size=9),
                                          transforms.Normalize((0.5,), (0.5,))
                                         ])
```

We also utilize in-batch negative examples according to (Chen et al., 2020). We implemented and trained using PyTorch (Paszke et al., 2019) on an A5000 GPU with 24GB of memory. The training duration does not exceed one day for all four datasets. We employ the Adam optimizer (Kingma & Ba, 2014) with initial learning rates ranging from 10^{-5} to 10^{-3}, depending on the dataset.

From preliminary results, we observed that a smaller value for K is inadequate for effectively learning significant image concepts. Conversely, a larger value for K tends to lead to redundancy among the population of all concepts. Consequently, we adhered to the baseline methodologies (e.g., (Fel et al., 2023b)) by setting K to 100 across all datasets. This number was chosen as it strikes an effective balance between capturing adequate detail and avoiding model overfitting.

Baselines Methods. For the implementation of the baseline methods, we either utilize the original packages provided by the authors (Lundberg & Lee, 2017; Ribeiro et al., 2016; Fel et al., 2023b), or implement their methods by referencing the authors' code (Simonyan et al., 2013; Pan et al., 2021). To account for the stochastic nature of methods like those in (Ribeiro et al., 2016; Lundberg & Lee, 2017), we perform multiple executions of these baseline methods, averaging scores across all runs. The frequency of repetition is contingent upon the time required to generate explanations. To balance efficiency and effectiveness, SHAP is executed 100 times, and LIME 10 times. In contrast, our PACE, being fully deterministic post-training, requires only a single inference on the test set.

Details on the Quantitative Analysis. In the results shown in Table 1 in Sec. 4.4, PACE offers *Full* three-level conceptual explanations, encompassing dataset, image, and patch levels. In contrast, CRAFT (Fel et al., 2023b) is limited to providing explanations at only the patch and image levels, lacking dataset-level insights and thereby exhibiting a *Partial* multi-level structure. Other baseline models are unable to achieve this multi-level conceptual explanation. Both the baseline CRAFT and our PACE are inherently compatible with arbitrary number of concepts K, therefore enjoying better parsimony by setting $K = 100$. This choice of K is driven by the goal of maintaining a moderate dimension size while ensuring that

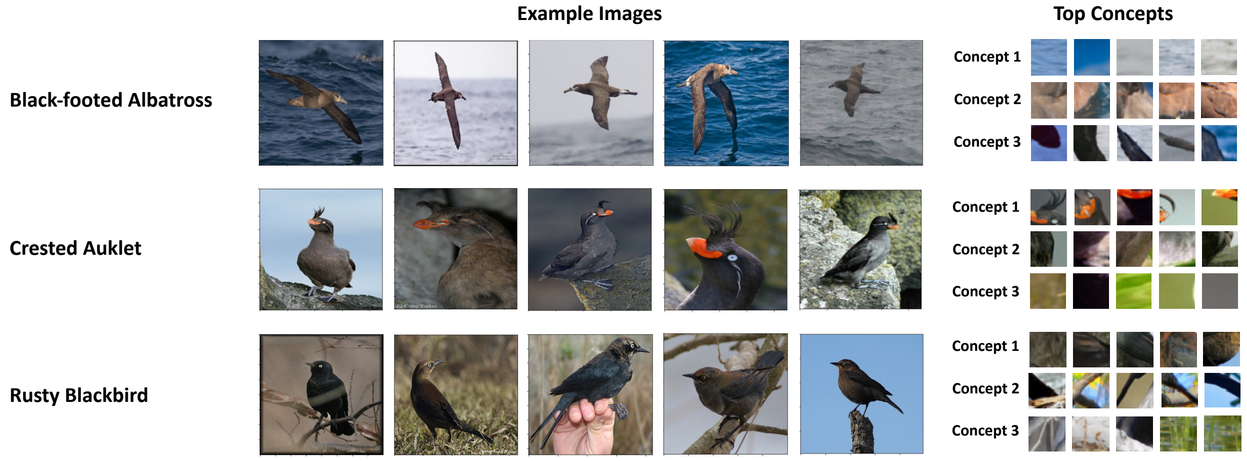

Figure 6. PACE's dataset-level conceptual explanations for classes **Black-footed Albatross**, **Crested Auklet**, and **Rusty Blackbird** in the *CUB* dataset. For each class, we show PACE's top 3 dataset-level concepts; for each Concept k, we show the top 5 patches with their associated embeddings $\mathbf{e}_{mj}$ closest to the concept center $\boldsymbol{\mu}_k$.

concept activation possesses meaningful semantics. In contrast, other baselines' number of concepts K is constrained to the hidden dimension of ViT embeddings, i.e., $K = 768$; they are therefore lacking in parsimony.

Details on the Qualitative Analysis. For qualitative analysis, we visualize 2×2 aggregated patches, chosen for their visibility and robustness against random noise. The aggregation only affects patch-level concepts, computed similarly to $\bar{\mathbf{z}}_m$ in Eq. 1. The mean of $\boldsymbol{\phi}_{mj}$ approximates the patch-level concept for each aggregated patch $\widehat{\boldsymbol{\phi}}_{mj}$, computed as follows:

$$\widehat{\boldsymbol{\phi}}_{m,u\cdot\frac{S}{2}+v} = \tfrac{1}{4}(\boldsymbol{\phi}_{m,2u\cdot S+2v} + \boldsymbol{\phi}_{m,2u\cdot S+2v+1} + \boldsymbol{\phi}_{m,(2u+1)\cdot S+2v} + \boldsymbol{\phi}_{m,(2u+1)\cdot S+2v+1}), \tag{15}$$

where S is the number of rows and columns. Note that this aggregation is for visualization purposes only during *qualitative* analysis and does not affect the quantitative results, dataset-level and image-level concepts, or the learning process.

B. Sparsity versus Parsimony

As we discuss in Definition 3.1, *sparsity* is defined as the proportion of $\boldsymbol{\theta}_m$'s entries nearing zero, i.e., $\frac{1}{K}\sum_{k=1}^{K} \mathbb{1}(|\boldsymbol{\theta}_{mk}| < \epsilon)$, with a small threshold $\epsilon > 0$; *parsimony* is defined as the minimal number of concepts K to produce clear and simple explanations. While sparsity is an *image-level* property, parsimony is *dataset-level*.

Example 1. For example, first consider a dataset with four concepts and three images, $\mathbf{I}_1$, $\mathbf{I}_2$, and $\mathbf{I}_3$:

- On the image level, $\mathbf{I}_1$: $\boldsymbol{\theta}_1 = (1, 0, 0, 0)$. $\mathbf{I}_2$: $\boldsymbol{\theta}_2 = (0, 1, 0, 0)$. $\mathbf{I}_3$: $\boldsymbol{\theta}_1 = (0, 0, 1, 0)$.
- On the dataset level, $\boldsymbol{\mu}_m = \mathbf{e}_m (1 \le m \le 3)$, $\boldsymbol{\mu}_4 = \frac{1}{3}(\boldsymbol{\mu}_1 + \boldsymbol{\mu}_2 + \boldsymbol{\mu}_3)$.

According to Definition 3.1, these image-level concepts satisfies *sparsity*; however, the dataset-level concept does not satisfy *parsimony*, since the last concept center $\boldsymbol{\mu}_4$ is redundant.

Example 2. Next, we consider a dataset with three concepts and three images, $\mathbf{I}_1$, $\mathbf{I}_2$, and $\mathbf{I}_3$:

- On the image level, $\mathbf{I}_1$: $\boldsymbol{\theta}_1 = (0.5, 0.5, 0)$. $\mathbf{I}_2$: $\boldsymbol{\theta}_2 = (0.5, 0, 0.5)$. $\mathbf{I}_3$: $\boldsymbol{\theta}_1 = (0, 0.5, 0.5)$.
- On the dataset level, $\boldsymbol{\mu}_1 = \frac{1}{2}(\mathbf{e}_1 + \mathbf{e}_2)$, $\boldsymbol{\mu}_2 = \frac{1}{2}(\mathbf{e}_1 + \mathbf{e}_3)$, and $\boldsymbol{\mu}_1 = \frac{1}{2}(\mathbf{e}_2 + \mathbf{e}_3)$.

According to Definition 3.1, These image-level concepts apparently does not satisfy *sparsity*; however, the dataset-level concept satisfies *parsimony*, because there are no redundant concepts. Therefore, in our paper, *sparsity and parsimony, though related, are distinct and non-interchangeable properties.*

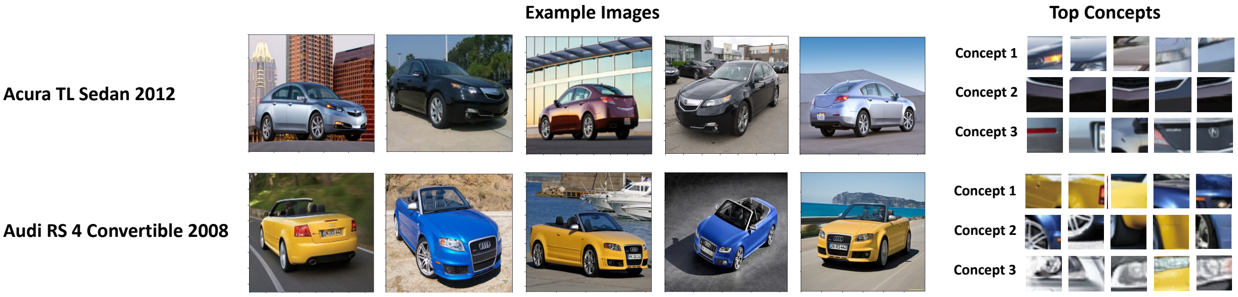

Figure 7. PACE's dataset-level conceptual explanations for classes **Acura TL Sedan 2012** and **Audi RS 4 Convertible 2008** in the *Cars* dataset. For each class, we show PACE's top 3 dataset-level concepts; for each Concept k, we show the top 5 patches with their associated embeddings $\mathbf{e}_{mj}$ closest to the concept center $\boldsymbol{\mu}_k$.

C. More Qualitative Results

In Fig. 6 and Fig. 7, we present the top three concepts for several distinct classes in the *CUB* and *Cars* datasets, respectively. Each concept is illustrated with the top five patches, providing dataset-level explanations.

Results on *CUB*. Fig. 6 shows PACE's dataset-level conceptual explanations for the *CUB* dataset's classes **Black-footed Albatross**, **Crested Auklet**, and **Rusty Blackbird**. For instance, the class **Black-footed Albatross** encompasses three predominant concepts: Concept 1 (*Ocean Background*), Concept 2 (*Brown Feather*), and Concept 3 (*Long Wing*). The accompanying top five patches exemplify PACE's conceptual explanations, highlighting critical dataset-level concepts such as the habitat (*Ocean*), distinctive texture (*Brown Feather*), and characteristic posture (*Long Wing*) crucial for classifying **Black-footed Albatross**. Similarly, the class **Crested Auklet** is distinguished by concepts such as Concept 1 (*Orange Beak*), Concept 2 (*Grey Feather*), and Concept 3 (*Rocks/Moss*); similarly, the class **Rusty Blackbird** is distinguished by Concept 1 (*Rusty Feather*), Concept 2 (*Tail*), and Concept 3 (*Grass/Branch*). These findings reveal that distinct bird classes are each linked to unique body characteristics, such as color, shape, and texture, as well as specific habitats.

Results on *Cars*. Fig. 7 shows PACE's dataset-level conceptual explanations for the *Cars* dataset's classes, such as **Acura TL Sedan 2012** and **Audi RS 4 Convertible 2008**. For example, the **Audi RS 4 Convertible 2008** class features three prominent concepts: Concept 1 (*Front Light*), Concept 2 (*Grill*), and Concept 3 (*Rear*). The top five patches representing these concepts indicate that design elements like the front light (*Front Light*), grill pattern (*Grill*), and rear features (*Rear*) are essential for classifying an image as class **Acura TL Sedan 2012**. Moreover, the **Audi RS 4 Convertible 2008** class is defined by concepts such as Concept 1 (*Streamline*), Concept 2 (*Tire/Fender*), and Concept 3 (*Front Light*), suggesting that car classes differ in design aspects such as shape and color.

Remark. In summary, these results showcase PACE's proficiency in identifying crucial dataset-level concepts across different classes, utilizing patch representations within the ViT framework. Notably, this process, once the model is trained, involves deducing top concepts for each class via inference, eliminating the need for retraining or finetuning. This approach is both efficient and effective compared to methods like CRAFT (Fel et al., 2023b) that require training for each individual class.

D. Details on Inferring Concepts

In this section, we discuss $g(\cdot)$ defined in Sec. 3.1 in detail.

In PACE, $g(\cdot)$ is implemented as an inference process on our Probabilistic Graphical Model (PGM), as shown in Fig. 2. One can see $g(\cdot)$ as a function that

(1) takes the observed variables $\mathbf{e}_m, \mathbf{e}'_m, \widehat{\mathbf{y}}_m, \mathbf{a}_m, \mathbf{a}'_m, y_s$ as inputs,
(2) goes through the learning stage and the inference stage discussed in Sec. 3.5 and Sec. 3.4, and
(3) outputs $\boldsymbol{\theta}_m$, the image-level concept explanations for each image m.

As shown in Fig. 1, $g(\cdot)$ refers to the PACE model (the gray box on the top right). It takes as inputs the patch embeddings $\mathbf{e}_m$, the attention weights $\mathbf{a}_m$ (which can be computed given the ViT's parameters P), and the ViT's predicted label $\widehat{\mathbf{y}}_m$; it then outputs the image-level explanations $\boldsymbol{\theta}$, i.e., $\boldsymbol{\theta}_m = g(\mathbf{e}_m, \widehat{\mathbf{y}}_m, \mathbf{a}_m)$.

Besides the image-level concept explanations $\boldsymbol{\theta}_m$, PACE also produces the dataset-level explanations $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ (where $k = 1, \ldots, K$) as well as patch-level explanations $\boldsymbol{\phi}_{mj}$ for patch j of image m).

$g(\cdot)$ is represented by the entire Fig. 2, except for the dashed box (with the text "ViT" inside). For example, during the inference stage, PACE will

(1) be given the *global* parameters $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ (where $k = 1, \ldots, K$) obtained from the learning stage,
(2) treats the patch embeddings $\mathbf{e}_m$, the attention weights $\mathbf{a}_m$ (which can be computed given the ViT's parameters P), and the ViT's predicted label $\widehat{\mathbf{y}}_m$ as observed variables,
(3) and then, for a new image m, infer the *local* parameters, i.e.,
    (a) the image-level concepts (explanations) $\boldsymbol{\theta}_m$, which is parameterized by $q(\boldsymbol{\theta}_m|\boldsymbol{\gamma}_m)$ and
    (b) patch-level concepts (explanations) $\mathbf{z}_{mj}$, which is parameterized by $q(\mathbf{z}_{mj}|\boldsymbol{\phi}_m)$.

These are called *local* parameters because each image has its own $\boldsymbol{\theta}_m$ and $\boldsymbol{\phi}_m$.

E. Theoretical Analysis

We provide the following proof of Theorem 3.2. For convenience, let $\Omega = (\boldsymbol{\mu}_{k=1}^K, \Sigma_{k=1}^K)$. We then introduce a helper joint distribution of the variables $\mathbf{e}_m$ and $\boldsymbol{\theta}_m, \boldsymbol{\phi}_m$, $s(\mathbf{e}_m, \boldsymbol{\theta}_m, \boldsymbol{\phi}_m) = p(\mathbf{e}_m) q(\boldsymbol{\theta}_m, \boldsymbol{\phi}_m|\mathbf{e}_m)$.

According to the definition of ELBO of Sec. 3.4, in Eq. 5 and Eq. 6, we only need to prove that

$$\begin{aligned} LHS = {} & L_e + L_f + L_s \\ \leq {} & I(\mathbf{e}_m; \boldsymbol{\theta}_m, \boldsymbol{\phi}_m) + I(\widehat{\mathbf{y}}_m; \boldsymbol{\phi}_m) + I(\boldsymbol{\phi}_m; \boldsymbol{\phi}'_m) + C. \end{aligned} \tag{16}$$

We split the proof into the following three separate part:

(1) The bound of L_e. We have that

$$L_e = \mathbb{E}_{p(\mathbf{e}_m)}[\mathbb{E}_q[\log p(\mathbf{e}_m|\Omega, \boldsymbol{\theta}_m, \boldsymbol{\phi}_m)]] + \mathbb{E}_q[\log q(\boldsymbol{\theta}_m, \boldsymbol{\phi}_m|\Omega)]. \tag{17}$$

Since $\mathbb{E}_q[\log q(\boldsymbol{\theta}_m, \boldsymbol{\phi}_m|\Omega)] \leq 0$, we are going to prove that

$$L_e \leq \mathbb{E}_{p(\mathbf{e}_m)}[\mathbb{E}_q[\log p(\mathbf{e}_m|\Omega, \boldsymbol{\theta}_m, \boldsymbol{\phi}_m)]] \leq I_s(\mathbf{e}_m; \boldsymbol{\theta}_m, \boldsymbol{\phi}_m) - H(\mathbf{e}_m). \tag{18}$$

In fact,

$$\begin{aligned} \mathbb{E}_{p(\mathbf{e}_m)}[\mathbb{E}_q[\log p(\mathbf{e}_m|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m, \Omega)]] & \leq \mathbb{E}_{p(\mathbf{e}_m)}[\mathbb{E}_q[\log p(\mathbf{e}_m|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m)]] \\ & = \mathbb{E}_{p(\mathbf{e}_m)}[\mathbb{E}_q[\log \tfrac{q(\mathbf{e}_m|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m)}{p(\mathbf{e}_m)} \tfrac{p(\mathbf{e}_m)p(\mathbf{e}_m|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m)}{q(\mathbf{e}_m|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m)}]] \\ & = \mathbb{E}_{p(\mathbf{e}_m)}[\mathbb{E}_q[\log \tfrac{q(\mathbf{e}_m|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m)}{p(\mathbf{e}_m)}]] + \mathbb{E}_{p(\mathbf{e}_m)}[\mathbb{E}_q[\log p(\mathbf{e}_m)]] + \mathbb{E}_{p(\mathbf{e}_m)}[\mathbb{E}_q[\log \tfrac{p(\mathbf{e}_m|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m)}{q(\mathbf{e}_m|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m)}]] \\ & = I_s(\mathbf{e}_m; \boldsymbol{\theta}_m, \boldsymbol{\phi}_m) - H(\mathbf{e}_m) - \mathbb{E}_q[KL(q(\mathbf{e}_m|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m)|p(\mathbf{e}_m|\boldsymbol{\theta}_m, \boldsymbol{\phi}_m))] \\ & \leq I_s(\mathbf{e}_m; \boldsymbol{\theta}_m, \boldsymbol{\phi}_m) - H(\mathbf{e}_m) - 0, \end{aligned} \tag{19}$$

where $H(\mathbf{e}_m)$ is a constant.

(2) The bound of L_f. With the constraint $-\mathbf{1} \leq \boldsymbol{\eta}_n \leq \mathbf{1} (1 \leq n \leq N)$, we have that

$$\begin{aligned} L_f = {} & \mathbb{E}_q[\log p(\widehat{\mathbf{y}}_m|\bar{\mathbf{z}}_m, \mathbf{H})] \\ \approx {} & \sum\nolimits_{n=1}^{N} \widehat{y}_{mn}(\boldsymbol{\eta}_n^T \bar{\boldsymbol{\phi}}_m) - \log(\sum\nolimits_{n=1}^{N} \exp(\boldsymbol{\eta}_n^T \bar{\boldsymbol{\phi}}_m)) \\ \leq {} & \sum\nolimits_{n=1}^{N} \widehat{y}_{mn}(\boldsymbol{\eta}_n^T \bar{\boldsymbol{\phi}}_m) \\ \leq {} & \sum\nolimits_{n=1}^{N} \sum\nolimits_{k=1}^{K} \widehat{y}_{mn} \bar{\phi}_{mk} \\ \leq {} & \sum_{\widehat{\mathbf{y}}_m} \sum_{\boldsymbol{\phi}_m} p(\widehat{\mathbf{y}}_m, \bar{\boldsymbol{\phi}}_m) \log \tfrac{p(\widehat{\mathbf{y}}_m, \bar{\boldsymbol{\phi}}_m)}{p(\widehat{\mathbf{y}}_m) p(\bar{\boldsymbol{\phi}}_m)} + C_1 \\ = {} & I(\widehat{\mathbf{y}}_m; \boldsymbol{\phi}_m) + C_1, \end{aligned} \tag{20}$$

where C_1 is a constant.

(3) The bound of L_s. With the constraint $\mathbf{0} \leq \boldsymbol{\beta} \leq \mathbf{1}$, we have that

$$\begin{aligned} L_s = \; & \mathbb{E}_q[\log p(y_s = 1 | \bar{\mathbf{z}}_{1:M}, \bar{\mathbf{z}}'_m, \boldsymbol{\beta})] \\ \approx \; & \boldsymbol{\beta}^T(\bar{\boldsymbol{\phi}}_m \circ \bar{\boldsymbol{\phi}}'_m) - \log(\sum\nolimits_{f \in \mathcal{F}} \exp(\boldsymbol{\beta}^T(\bar{\boldsymbol{\phi}}_m \circ \bar{\boldsymbol{\phi}}_f))) \\ \leq \; & \boldsymbol{\beta}^T(\bar{\boldsymbol{\phi}}_m \circ \bar{\boldsymbol{\phi}}'_m) \\ \leq \; & \bar{\boldsymbol{\phi}}_m \cdot \bar{\boldsymbol{\phi}}'_m \\ \leq \; & \sum_{\boldsymbol{\phi}_m} \sum_{\boldsymbol{\phi}'_m} p(\boldsymbol{\phi}_m, \boldsymbol{\phi}'_m) \log \tfrac{p(\boldsymbol{\phi}_m, \boldsymbol{\phi}'_m)}{p(\boldsymbol{\phi}_m) p(\boldsymbol{\phi}'_m)} + C_2 \\ = \; & I(\boldsymbol{\phi}_m; \boldsymbol{\phi}'_m) + C_2, \end{aligned} \tag{21}$$

where C_2 is a constant.

Combining $(1 \sim 3)$ above concludes the proof.

F. Details on Learning PACE

F.1. Derivations of ELBO

Inferring $\mathbf{z}_m$. According to Eq. 1,

$$\bar{\mathbf{z}}_m = \tfrac{1}{J} \sum\nolimits_{j=1}^{J} \mathbf{z}_{mj}, \tag{22}$$

where $\mathbf{z}_{mj}$ can be approximate by a variational distribution parameterized by $\boldsymbol{\phi}_{mj}$:

$$q(\mathbf{z}_{mj} \mid \boldsymbol{\phi}_{mj}) = \text{Categorical}(\mathbf{z}_{mj} \mid \boldsymbol{\phi}_{mj}), \tag{23}$$

which indicates that

$$\mathbb{E}[\mathbf{z}_{mj}] = \boldsymbol{\phi}_{mj}. \tag{24}$$

Therefore, we have

$$\mathbb{E}[\bar{\mathbf{z}}_m] = \tfrac{1}{J} \sum\nolimits_{j=1}^{J} \mathbb{E}[\bar{\mathbf{z}}_{mj}] = \tfrac{1}{J} \sum\nolimits_{j=1}^{J} \boldsymbol{\phi}_{mj} = \bar{\boldsymbol{\phi}}_m. \tag{25}$$

Hence, we have

$$\bar{\mathbf{z}}_m \approx \bar{\boldsymbol{\phi}}_m. \tag{26}$$

Computing L_e. We can expand the ELBO in Eq. 7 as:

$$\begin{aligned} L_e = \; & \log \boldsymbol{\Gamma}(\sum_{k=1}^{K} \alpha_k) - \sum_{k=1}^{K} \log \boldsymbol{\Gamma}(\alpha_k) + \sum_{k=1}^{K} (\alpha_k - 1)(\Psi(\boldsymbol{\gamma}_{mk}) - \Psi(\sum_{k'=1}^{K} \boldsymbol{\gamma}_{k'})) \\ & + \sum_{k=1}^{K} \phi_{mjk} (\Psi(\boldsymbol{\gamma}_{mk}) - \Psi(\sum_{k'=1}^{K} \boldsymbol{\gamma}_{mk'})) \\ & + \sum_{k=1}^{K} \phi_{mjk} a_{mj} \{ -\tfrac{1}{2} (\mathbf{e}_{mj} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\mathbf{e}_{mj} - \boldsymbol{\mu}_k) - \log[(2\pi)^{d/2} |\boldsymbol{\Sigma}_k|^{1/2}] \} \\ & - \log \boldsymbol{\Gamma}(\sum_{k=1}^{K} \boldsymbol{\gamma}_{mk}) + \sum_{k=1}^{K} \log \boldsymbol{\Gamma}(\boldsymbol{\gamma}_{mk}) - \sum_{k=1}^{K} (\boldsymbol{\gamma}_{mk} - 1)(\Psi(\boldsymbol{\gamma}_{mk}) - \Psi(\sum_{k'=1}^{K} \boldsymbol{\gamma}_{mk'})) \\ & - \sum_{k=1}^{K} \phi_{mjk} \log \phi_{mjk}. \end{aligned} \tag{27}$$

We can interpret the meaning of each term of L_e as follows:

- The sum of the first and the fourth terms, namely $\mathbb{E}_q[\log p(\boldsymbol{\theta}_m|\boldsymbol{\alpha})] - \mathbb{E}_q[\log q(\boldsymbol{\theta}_m)]$, is equal to $-KL(q(\boldsymbol{\theta}_m)|p(\boldsymbol{\theta}_m|\boldsymbol{\alpha}))$, which is the negation of KL Divergence between the variational posterior probability $q(\boldsymbol{\theta}_m)$ and the prior probability $p(\boldsymbol{\theta}_m|\boldsymbol{\alpha})$ of the topic proportion $\boldsymbol{\theta}_m$ for document m. Therefore maximizing the sum of these two terms is equivalent to minimizing the KL Divergence $KL(q(\boldsymbol{\theta}_m)|p(\boldsymbol{\theta}_m|\boldsymbol{\alpha}))$; this serves as a regularization term to make sure the inferred $q(\boldsymbol{\theta}_m)$ is close to its prior distribution $p(\boldsymbol{\theta}_m|\boldsymbol{\alpha})$.
- Similarly, the sum of the second and the last terms (ignoring the summation over the word index j for simplicity), namely $\mathbb{E}_q[\log p(z_{mj}|\boldsymbol{\theta}_m)] - \mathbb{E}_q[\log q(z_{mj})]$ is equal to $-KL(q(z_{mj})|p(z_{mj}|\boldsymbol{\theta}_m))$, which is the negation of the KL Divergence between the variational posterior probability $q(z_{mj})$ and the prior probability $p(z_{mj}|\boldsymbol{\theta}_m)$ of the word-level topic assignment z_{mj} for word j of document m. Therefore maximizing the sum of these two terms is equivalent to minimizing the KL Divergence $KL(q(z_{mj})|p(z_{mj}|\boldsymbol{\theta}_m))$; this serves as a regularization term to make sure the inferred $q(z_{mj})$ is close to its "prior" distribution $p(z_{mj}|\boldsymbol{\theta}_m)$.
- The third term $\mathbb{E}_q[\log p(\mathbf{e}_{mj}|z_{mj}, \boldsymbol{\mu}_{z_{mj}}, \boldsymbol{\Sigma}_{z_{mj}})]$ is to maximize the log likelihood $p(\mathbf{e}_{mj}|z_{mj}, \boldsymbol{\mu}_{z_{mj}}, \boldsymbol{\Sigma}_{z_{mj}})$ of every contextual embedding $\mathbf{e}_{mj}$ (for word j of document m) conditioned on the inferred z_{mj} and the parameters $(\boldsymbol{\mu}_{z_{mj}}, \boldsymbol{\Sigma}_{z_{mj}})$.

Computing L_f. Eq. 8 is derived from employing Taylor's expansion to Eq. 2:

$$\begin{aligned} L_f &= \mathbb{E}_q[\log p(\widehat{\mathbf{y}}_m|\bar{\mathbf{z}}_m, \mathbf{H})] \\ &= \sum\nolimits_{n=1}^{N} \widehat{y}_{mn}(\boldsymbol{\eta}_n^T \bar{\boldsymbol{\phi}}_m) - \mathbb{E}_q[\log(\sum\nolimits_{n=1}^{N} \exp(\boldsymbol{\eta}_n^T \bar{\mathbf{z}}_m))] \\ &\approx \sum\nolimits_{n=1}^{N} \widehat{y}_{mn}(\boldsymbol{\eta}_n^T \bar{\boldsymbol{\phi}}_m) - \log(\sum\nolimits_{n=1}^{N} \exp(\boldsymbol{\eta}_n^T \bar{\boldsymbol{\phi}}_m + (1/2)\boldsymbol{\eta}_n^T \mathbf{S}_m \boldsymbol{\eta}_n)), \end{aligned} \tag{28}$$

where $\mathbf{S}_m$ is the covariance matrix of $\bar{\mathbf{z}}_m$.

We will see that for any entry of $\mathbf{S}_m$, i.e. $\forall x, y \in \{1, 2, ..., K\}$, we have

$$0 \le \mathbf{S}_{m,xy} \le \tfrac{1}{J^2}. \tag{29}$$

In our setting, the number of patches J in each image satisfies $J > 100$, hence $\mathbf{S}_{m,xy}$ is very close to zero.

We compute $\mathbf{S}_m$ by definition:

$$\begin{aligned} \mathbf{S}_{m,xy} &= Cov[\bar{\mathbf{z}}_m \bar{\mathbf{z}}_{m'}]_{x,y} \\ &= \mathbb{E}[(\bar{z}_{mx}\bar{z}_{m'x} - \mathbb{E}[\bar{z}_{mx}\bar{z}_{m'x}])(\bar{z}_{my}\bar{z}_{m'y} - \mathbb{E}[\bar{z}_{my}\bar{z}_{m'y}])] \\ &= \mathbb{E}[(\bar{z}_{mx}\bar{z}_{m'x} - \mathbb{E}[\bar{z}_{mx}]\mathbb{E}[\bar{z}_{m'x}])(\bar{z}_{my}\bar{z}_{m'y} - \mathbb{E}[\bar{z}_{my}]\mathbb{E}[\bar{z}_{m'y}])] \\ &= \mathbb{E}[(\bar{z}_{mx}\bar{z}_{m'x} - \bar{\phi}_{mx}\bar{\phi}_{m'x})(\bar{z}_{my}\bar{z}_{m'y} - \bar{\phi}_{my}\bar{\phi}_{m'y})] \\ &= \mathbb{E}[\bar{z}_{mx}\bar{z}_{m'x}\bar{z}_{my}\bar{z}_{m'y}] - \bar{\phi}_{my}\bar{\phi}_{m'y}\mathbb{E}[\bar{z}_{mx}\bar{z}_{m'x}] - \bar{\phi}_{mx}\bar{\phi}_{m'x}\mathbb{E}[\bar{z}_{my}\bar{z}_{m'y}] + \bar{\phi}_{mx}\bar{\phi}_{m'x}\bar{\phi}_{my}\bar{\phi}_{m'y} \\ &= \mathbb{E}[\bar{z}_{mx}\bar{z}_{my}]\mathbb{E}[\bar{z}_{m'x}\bar{z}_{m'y}] - \bar{\phi}_{mx}\bar{\phi}_{m'x}\bar{\phi}_{my}\bar{\phi}_{m'y}. \end{aligned} \tag{30}$$

We then consider two different cases:

Case (1): $x = y$**.**

Then we have that

$$\begin{aligned} Cov[\bar{\mathbf{z}}_m \bar{\mathbf{z}}_{m'}]_{x,y} &= \mathbb{E}[\bar{z}_{mx}^2]\mathbb{E}[\bar{z}_{m'y}^2] - \bar{\phi}_{mx}^2\bar{\phi}_{m'y}^2 \\ &= \bar{\phi}_{mx}^2\bar{\phi}_{m'y}^2 - \bar{\phi}_{mx}^2\bar{\phi}_{m'y}^2 \\ &= 0. \end{aligned} \tag{31}$$

Case (2): $x \neq y$**.**

Note that

$$\bar{z}_{mx}\bar{z}_{my} = \tfrac{1}{J}\sum_j z_{mjx} \cdot \tfrac{1}{J}\sum_j z_{mjy}. \tag{32}$$

Given that $\mathbf{z}_{mj}$ is a one-hot vector, we have

$$z_{mjx} \cdot z_{mjy} = 0. \tag{33}$$

Hence, we have

$$\begin{aligned} \mathbb{E}[\bar{z}_{mx}\bar{z}_{my}] =& \ \mathbb{E}[\bar{z}_{mx}]\mathbb{E}[\bar{z}_{my}] - \tfrac{1}{J^2}(\sum_{j=1}^{J} \mathbb{E}[z_{mjx}]\mathbb{E}[z_{mjy}]) \\ =& \ \bar{\phi}_{mx}\bar{\phi}_{my} - \tfrac{1}{J^2}\sum_{j=1}^{J} \phi_{mjx}\phi_{mjy}. \end{aligned} \tag{34}$$

Therefore, we have

$$\begin{aligned} Cov[\bar{\mathbf{z}}_m\bar{\mathbf{z}}_{m'}]_{x,y} =& \ (\bar{\phi}_{mx}\bar{\phi}_{my} - \tfrac{1}{J^2}\sum_{j=1}^{J} \phi_{mjx}\phi_{mjy})(\bar{\phi}_{m'x}\bar{\phi}_{m'y} - \tfrac{1}{J^2}\sum_{j=1}^{J} \phi_{m'jx}\phi_{m'jy}) - \bar{\phi}_{mx}\bar{\phi}_{m'x}\bar{\phi}_{my}\bar{\phi}_{m'y} \\ =& \ \tfrac{1}{J^4}\sum_{j=1}^{J} \phi_{mjx}\phi_{mjy}\sum_{j=1}^{J} \phi_{m'jx}\phi_{m'jy}. \end{aligned} \tag{35}$$

Since $\mathbf{0} \leq \boldsymbol{\phi}_{mj}, \boldsymbol{\phi}'_{mj} \leq \mathbf{1}$, we have that

$$0 \leq \mathbf{S}_{m,xy} = Cov[\bar{\mathbf{z}}_m\bar{\mathbf{z}}_{m'}]_{x,y} \leq \tfrac{1}{J^2}. \tag{36}$$

In summary, we can see that in either case, Eq. 29 holds.

Therefore we have

$$L_f \approx \sum\nolimits_{n=1}^{N} \widehat{y}_{mn}(\boldsymbol{\eta}_n^T\bar{\boldsymbol{\phi}}_m) - \log(\sum\nolimits_{n=1}^{N} \exp(\boldsymbol{\eta}_n^T\bar{\boldsymbol{\phi}}_m)). \tag{37}$$

Computing L_s. Similarly, by employ Taylor's expansion of Eq. 3, as well as Eq. 29, we have that

$$\begin{aligned} L_s =& \ \mathbb{E}_q[\log p(y_s = 1 | \bar{\mathbf{z}}_{1:M}, \bar{\mathbf{z}}'_m, \boldsymbol{\beta})] \\ =& \ \boldsymbol{\beta}^T(\bar{\mathbf{z}}_m \circ \bar{\mathbf{z}}'_m) - \mathbb{E}_q[\log(\sum\nolimits_{f\in\mathcal{F}} \exp(\boldsymbol{\beta}^T(\bar{\mathbf{z}}_m \circ \bar{\mathbf{z}}_f)))] \\ \approx& \ \boldsymbol{\beta}^T\bar{\boldsymbol{\phi}}_m\bar{\boldsymbol{\phi}}'_m - \log(\sum\nolimits_{f\in\mathcal{F}} \exp(\boldsymbol{\beta}^T(\bar{\boldsymbol{\phi}}_m\bar{\boldsymbol{\phi}}_f) + (1/2)\boldsymbol{\beta}^T\mathbf{S}_m\boldsymbol{\beta})) \\ \approx& \ \boldsymbol{\beta}^T(\bar{\boldsymbol{\phi}}_m \circ \bar{\boldsymbol{\phi}}'_m) - \log(\sum\nolimits_{f\in\mathcal{F}} \exp(\boldsymbol{\beta}^T(\bar{\boldsymbol{\phi}}_m \circ \bar{\boldsymbol{\phi}}_f))), \end{aligned} \tag{38}$$

where the parameter $\boldsymbol{\beta}$ is learned jointly by gradient-based optimization algorithms, such as Adam.

F.2. Update Rules

F.2.1. Inference

Derivative of L_e. Taking the derivative of the L_e in Eq. 7 with respect to ϕ_{mjk} and setting it to zero, we obtain the update rule for ϕ_{mjk}:

$$\begin{aligned} \phi_{mjk} \propto& \ \tfrac{1}{|\boldsymbol{\Sigma}_k|^{1/2}} \exp[\Psi(\gamma_{mk}) - \Psi(\sum\nolimits_{k'=1}^{K} \gamma_{k'}) \\ & -\tfrac{1}{2}a_{mj}(\mathbf{e}_{mj} - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{e}_{mj} - \boldsymbol{\mu}_k)]. \end{aligned} \tag{39}$$

Derivative of L_f. The log-sum term of Eq. 37 is intractable. To address this, with Taylor's Expansion, we have that

$$\log(\sum\nolimits_{n=1}^{N} \exp(\boldsymbol{\eta}_n^T\bar{\boldsymbol{\phi}}_m)) \approx \ \log(\sum\nolimits_{n=1}^{N} \exp(\boldsymbol{\eta}_n^T\bar{\boldsymbol{\phi}}_m^{(0)})) + (\bar{\boldsymbol{\phi}}_m - \bar{\boldsymbol{\phi}}_m^{(0)})^T \tfrac{\sum_{n=1}^{N} \exp(\boldsymbol{\eta}_n^T\bar{\boldsymbol{\phi}}_m^{(0)})\boldsymbol{\eta}_n}{\sum_{n=1}^{N} \exp(\boldsymbol{\eta}_n^T\bar{\boldsymbol{\phi}}_m^{(0)})}, \tag{40}$$

where $\bar{\boldsymbol{\phi}}_m^{(0)}$ is the value of $\bar{\boldsymbol{\phi}}_m$ at the last iteration of $\boldsymbol{\phi} - \boldsymbol{\gamma}$ update discussed in Alg. 1.

Taking the derivative w.r.t. $\boldsymbol{\phi}_m$, we have that

$$\frac{\partial L_f}{\partial \bar{\boldsymbol{\phi}}_m} \approx \sum_{n=1}^{N} y_{mn}\boldsymbol{\eta}_n - \frac{\sum_{n=1}^{N}\exp(\boldsymbol{\eta}_n^T\bar{\boldsymbol{\phi}}_m)\boldsymbol{\eta}_n}{\sum_{n=1}^{N}\exp(\boldsymbol{\eta}_n^T\bar{\boldsymbol{\phi}}_m)}. \tag{41}$$

Note that by definition $\bar{\boldsymbol{\phi}}_m = 1/J\sum_{j=1}^{J}\boldsymbol{\phi}_{mj}$ in the main paper, we have

$$\frac{\partial L_f}{\partial \boldsymbol{\phi}_{mj}} = \frac{\partial L_f}{\partial \bar{\boldsymbol{\phi}}_m} \cdot \frac{\partial \bar{\boldsymbol{\phi}}_m}{\partial \boldsymbol{\phi}_{mj}} = \frac{1}{N}\frac{\partial L_f}{\partial \bar{\boldsymbol{\phi}}_m} = \frac{1}{N}(\sum_{n=1}^{N} y_{mn}\boldsymbol{\eta}_n - \frac{\sum_{n=1}^{N}\exp(\boldsymbol{\eta}_n^T\bar{\boldsymbol{\phi}}_m)\boldsymbol{\eta}_n}{\sum_{n=1}^{N}\exp(\boldsymbol{\eta}_n^T\bar{\boldsymbol{\phi}}_m)}). \tag{42}$$

Derivative of L_s. Taking the derivative of Eq. 38, i.e.

$$L_s = \boldsymbol{\beta}^T(\bar{\boldsymbol{\phi}}_m \circ \bar{\boldsymbol{\phi}}_m') - \log(\sum\nolimits_{f\in\mathcal{F}}\exp(\boldsymbol{\beta}^T(\bar{\boldsymbol{\phi}}_m \circ \bar{\boldsymbol{\phi}}_f))), \tag{43}$$

we have that

$$\frac{\partial L_s}{\partial \bar{\boldsymbol{\phi}}_m} \approx \boldsymbol{\beta}^T\bar{\boldsymbol{\phi}}_m' - \frac{\sum_{f\in\mathcal{F}}\exp(\boldsymbol{\beta}^T(\bar{\boldsymbol{\phi}}_m \circ \bar{\boldsymbol{\phi}}_f))\boldsymbol{\beta}^T\bar{\boldsymbol{\phi}}_f}{\sum_{f\in\mathcal{F}}\exp(\boldsymbol{\beta}^T(\bar{\boldsymbol{\phi}}_m \circ \bar{\boldsymbol{\phi}}_f))}. \tag{44}$$

In summary, the partial derivative of ELBO in Eq. 5 w.r.t. $\boldsymbol{\phi}_{mj}$ is

$$\frac{\partial L}{\partial \boldsymbol{\phi}_{mj}} = \frac{\partial L_e}{\partial \boldsymbol{\phi}_{mj}} + \frac{\partial L_f}{\partial \boldsymbol{\phi}_{mj}} + \frac{\partial L_s}{\partial \boldsymbol{\phi}_{mj}}. \tag{45}$$

Setting the derivative to 0, we have a closed-form update rules for $\boldsymbol{\phi}$ as follows:

$$\begin{aligned}\boldsymbol{\phi}_{mj} \propto\ & \frac{1}{|\boldsymbol{\Sigma}_k|^{1/2}}\exp[\Psi(\gamma_{mk}) - \Psi(\sum\nolimits_{k'=1}^{K}\gamma_{k'}) - \frac{1}{2}a_{mj}(\mathbf{e}_{mj} - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\mathbf{e}_{mj} - \boldsymbol{\mu}_k) \\ & + \frac{1}{J}(\sum\nolimits_{n=1}^{N}\widehat{y}_{ml}\boldsymbol{\eta}_l - \frac{\sum_{n'=1}^{N}\exp(\boldsymbol{\eta}_{n'}^T\bar{\boldsymbol{\phi}}_m)\boldsymbol{\eta}_n}{\sum_{n=1}^{N}\exp(\boldsymbol{\eta}_n^T\bar{\boldsymbol{\phi}}_m)} + \boldsymbol{\beta}^T\bar{\boldsymbol{\phi}}_m' - \frac{\sum_{f\in\mathcal{F}}\exp(\boldsymbol{\beta}^T(\bar{\boldsymbol{\phi}}_m \circ \bar{\boldsymbol{\phi}}_f))(\boldsymbol{\beta}^T\bar{\boldsymbol{\phi}}_f)}{\sum_{f\in\mathcal{F}}\exp(\boldsymbol{\beta}^T(\bar{\boldsymbol{\phi}}_m \circ \bar{\boldsymbol{\phi}}_f))})].\end{aligned} \tag{46}$$

Taking derivative of Eq. 27 and set to $\mathbf{0}$, we have

$$\gamma_{mk} = \alpha_k + \sum\nolimits_{j=1}^{J}\phi_{mjk}a_{mj}. \tag{47}$$

F.2.2. Learning

Similar to Sec. F.2.1, we expand the ELBO in Eq. 7, take its derivative w.r.t. $\boldsymbol{\mu}_k$ and set it to 0:

$$\frac{\partial L}{\partial \boldsymbol{\mu}_k} = \sum_{m,j}\phi_{mjk}a_{mj}\boldsymbol{\Sigma}_k^{-1}(\mathbf{e}_{mj} - \boldsymbol{\mu}_k) = 0, \tag{48}$$

yielding the update rule for learning $\boldsymbol{\mu}_k$:

$$\boldsymbol{\mu}_k = \frac{\sum_{m,j}\phi_{mjk}a_{mj}\mathbf{e}_{mj}}{\sum_{m,j}\phi_{mjk}a_{mj}}, \tag{49}$$

where $\boldsymbol{\Sigma}_k^{-1}$ is canceled out.

Similarly, setting the derivatives w.r.t. $\boldsymbol{\Sigma}_k$ to $\mathbf{0}$, i.e.,

$$\frac{\partial L}{\partial \boldsymbol{\Sigma}_k} = \frac{1}{2}\sum_{m,j}\phi_{mjk}a_{mj}(-\boldsymbol{\Sigma}_k^{-1} + \boldsymbol{\Sigma}_k^{-1}(\mathbf{e}_{mj} - \boldsymbol{\mu}_k)(\mathbf{e}_{mj} - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}), \tag{50}$$

we have

$$\boldsymbol{\Sigma}_k = \frac{\sum_{m,j}\phi_{mjk}a_{mj}(\mathbf{e}_{mj} - \boldsymbol{\mu}_k)(\mathbf{e}_{mj} - \boldsymbol{\mu}_k)^T}{\sum_{m,j}\phi_{mjk}a_{mj}}. \tag{51}$$