Universiteit Leiden
The Netherlands

Using deep learning models in image processing

Xiong, Z.

Citation
Xiong, Z. (2026, May 19). *Using deep learning models in image processing*. Retrieved from https://hdl.handle.net/1887/4303646

Version:	Publisher's Version
License:	Licence agreement concerning inclusion of doctoral thesis in the Institutional Repository of the University of Leiden
Downloaded from:	https://hdl.handle.net/1887/4303646

Note: To cite this publication please use the final published version (if applicable).

Chapter 3

VGMEA: Interpretable Cross-Attention mechanisms in Semantic Segmentation via Variational Gaussian Model Mixture

Preparing for arXiv pre-print.

Chapter Summary

The rise of attention mechanisms in computer vision has improved state-of-the-art performance and advanced network design. Originating in Natural Language Processing (NLP), attention mechanisms, including self-attention and cross-attention, effectively capture long-range dependencies between pairs of elements without regard to distance. Generally speaking, most existing attention modules can be roughly reformulated as either K-Means or K-NN clustering. K-Means style attention, maintaining clustering centers as global queries, favors building explicit feature spaces for closed-form vision tasks (e.g., semantic segmentation). In contrast, K-NN style attention, by generating dispensable clustering centers as dynamic queries, facilitates implicit feature spaces for open-form vision tasks (e.g., object detection and instance segmentation). In this chapter, we propose a variational Gaussian model mixture embedded attention (VGMEA) module to interpret cross-attention as a generalized probabilistic K-Means clustering approach. The VGMEA module also integrates geometrical regularization into cross-attention to improve the cross-class decision boundary based on non-local spatial dependencies.

Keyword: Probabilistic Inference Variational Bayes Attention Mechanism K-means Clustering

3.1 Introduction

The ability to effectively capture *long-range* dependencies across multiple scales is considered a crucial factor in improving the performance of deep neural networks across various tasks, from academia to industry. Traditionally, recurrent operations [558, 205] are the most widely used approach for modeling long-range dependencies in sequential data (e.g., speech and language). In contrast, large effective receptive fields formed by deep stacks of convolutional operations [186, 558] are the dominant approach to modeling long-distance interactions for spatial and temporal-spatial data (e.g., images and videos). However, a single recurrent or convolutional operation can only capture short-term relationships within a local neighborhood, either spatially or temporally. Therefore, capturing long-range dependencies can be approximated only by a chain of repeated operations, in which short-range signals are progressively propagated through each node.

Apparently, a long chain of repeating local operations has several limitations. First, approximating one operation through a chain will cause optimization instability that needs to be carefully addressed by some tricks, including dropout [258], residual shortcut [248]. Second, propagating signals step-wise along the chain is computationally inefficient. Finally, their perplexing computational graphs are too heavily weighted to be used as modules in other networks to model multi-scale dependencies, e.g., when messages must be communicated back and forth between distant positions at different scales. Therefore, attention mechanisms are operations designed to alleviate the aforementioned limitations. Those operators are more efficient in capturing long-range dependencies between pairwise positions.

Attention mechanisms have been a key component widely applied to improve the performance of deep neural networks on challenging visual datasets [138, 166, 416, 849, 123]. According to different purposes [214], attention mechanisms can be classified as channel attention [268, 665, 528], spatial attention [806, 53, 156], temporal attention [749, 5] and branch attention [97, 807]. Among these variants, some researchers [496, 44] explored the unified patterns to guide new attention mechanisms. All this work supports the idea that attention can enhance informative features, suppress irrelevant ones, and improve the learned feature representation by precisely modeling correlations between pairwise positions.

Compared to the significant enthusiasm for designing attention mechanisms, only a few works focus on their interpretability. Some relevant work [678, 379, 715, 773, 663] has partially explained why attention modules can highlight informative channels

and/or regions. However, there is still no systematic framework for categorizing these variants. The attention mechanism can be revisited as mapping a *query* and a set of *key-value* pairs to an output. The output is computed as a weighted sum of the *values*, where the weight of each *value* is computed by a metric function between the *query* and the corresponding *key*. If regarding the *query* as a cluster center, then the attention mechanism can be understood as computing distances from the center to corresponding *keys* in one vector space while outputting a center of *values* in another vector space. However, allowing *keys* and *values* in different feature spaces actually hinders a systematic top-down viewpoint from interpreting all attention variants. Considering *values* are linear transformations from *keys* (as most variants adopted), we make a rational constraint that *values* would be mapped onto the same vector space as *keys*. Therefore, an attention operation can be analogized as one step of updating/shifting the **old** center (i.e., *query*) to a **new** center, depicted in Figure 3.1. Hereafter, one attention module, i.e., a stack of attention operations, can be interpreted as a clustering process with corresponding steps.

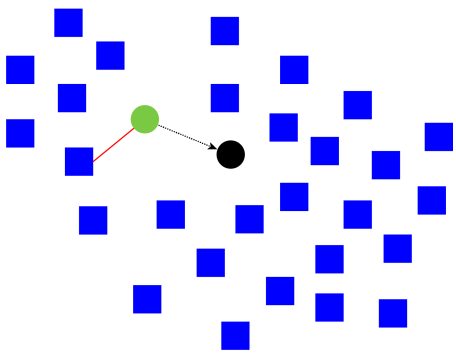


Figure 3.1: It is a straightforward visualization of an attention mechanism that maps a query and a set of key-value pairs to an output. Blue squares denote the input key-value pairs. The green circle is the input query, while the black circle is the output. In addition, the red edge between a blue square and the green circle represents the vector distance. Therefore, the attention mechanism can be interpreted as shifting the green circle forward to the black circle determined by all pairwise distances.

The top-down viewpoint is beneficial for applications since it can provide theoretical evidence to select suitable modules for different scenarios. Depending on the implementation, attention modules can be reformulated as two clustering processes: K-means and k-nearest neighbors (k-NN). Assume that there is a feature space that contains all input feature vectors. The K-means style attention [806, 379, 784, 654, 105, 773] aims to capture globally shared information across all input feature vectors by intentionally

smoothing [1] out their local/individual details. Subsequently, the smoothed part of all feature vectors will span an embedded subspace, which can distill the common information that exists among those features. Therefore, we argue that K-means attention is a suitable tool for closed-form classification problems, e.g., semantic segmentation, because all common information can be captured by an adequate number of centers progressively during the training stage. In contrast, the k-NN style attention [642, 678, 66, 870, 663, 855] focuses on finding subtle differences among a set of feature vectors that are already close to each other. In other words, all old centers only compute vector distances against their corresponding k closest neighbors within relatively narrow ranges, and arbitrary new unique centers will be determined conditioned on the inputs. Therefore, we argue that k-NN attentions are efficient tools for non-closed-form classification problems, e.g., object detection/instance segmentation, since the number of objects can vary across inputs, while delicate details must be well distinguished between objects within the same classes.

In this chapter, we are mainly interested in K-means attention and focus on analyzing the learnt embedded feature spaces of deep learning models in semantic segmentation problems. We rethink the K-means attention modules from a variational Bayes (VB) perspective [629] and propose a novel attention module, namely *variational Gaussian mixture model embedded attention* (VGMEA). The graphical model of VGMEA, depicted in Fig. 3.2b, is a generalization of the classical Gaussian mixture model (GMM) [467] in computer vision (shown in Fig. 3.2a) since more regularization terms can be added through the variational parameters ϕ_n. Intuitively, the VGMEA is a flexible linear superposition of Gaussian components, aimed at providing a richer class of density models with latent variables. The latent variables can be interpreted as attention maps that compute the response at a position as a correlation with *all Gaussian centers*. The set of positions can be in space, time, or space-time, implying that our models apply to images, sequences, and video problems. Unlike the other attention modules that compute derivatives with respect to parameters for backpropagation, we apply the VB algorithm to update distribution parameters using a momentum-based moving average. In this setting, the VB algorithm encodes all the available information about the model parameters in their posterior distribution. Given the current parameters, VB minimizes the evidence lower bound (ELBO) to alternatively approximate the posterior distribution of latent variables and update variational parameters. After convergence, the output can be computed as the new centers weighted by the final attention maps.

[1] removing, so-called *denoising*, highly varying information, and keeping common features.

- First, we interpret attention mechanisms as shifting the old centers among *keys* toward new centers among *values*. Moreover, K-means attention modules are reformulated as structured probabilistic graphs that use variational Bayes algorithms to iteratively update their parameters. Our model can effectively distill global shared information from feature maps and communicate these messages back and forth between distant positions at different scales.

- Second, we build our K-means attention as a light-weighted module for deep neural networks, and set up a momentum-based moving average strategy for model parameters' maintenance and normalization. Complicated regularization terms, e.g., total variation, are incorporated to control message passing among features of different scales.

- Third, extensive experiments are conducted on four challenging semantic segmentation datasets, including PASCAL Context, Cityscapes, MS-COCO, and ADE20K, demonstrating that the proposed model can interpret many other state-of-the-art methods as specific instances. The proposed model is generic in multi-scale attention learning and achieves competitive performance compared with state-of-the-art methods.

(a) GMM **(b)** VGMEA

Figure 3.2: The directed acyclic graphs represent the mixture of Gaussian models. The box (plate) denotes a set of N i.i.d. observations and latent variables. Each node of x_n in blue is an observation. Meanwhile, $(\boldsymbol{\mu}, \mathbf{C})$ denotes a set of K model variables, where $\boldsymbol{\mu}$ denotes $\{\mu_k\}$ and $\mathbf{C}$ denotes $\{\mathbf{C}_k\}$. (a) Classical GMM corresponds to an observation with a latent variable z_n and a variational variable ϕ_n. z_n comprises a K binary vector that represents the correlations between position n and K centers. It is determined by a Dirichlet, ϕ_n is a K-Dimensional real vector on a decision surface that can integrate geometrical regularization. (b) Our VGMEA corresponds to an observation with a latent variable z_n and a variational variable ϕ_n. z_n comprises a K binary vector that represents the correlations between position n and K centers. Moreover, ϕ_n is a K real vector on a decision surface that can integrate geometrical regularization.

There are several advantages of using VGMEA models: (a) VGMEA can be easily combined as **non-parametric** modules in the end-to-end training of existing neural networks, due to self-updating parameters via the VB algorithm, adding no extra derivatives to the computational graphs; (b) since decoupled from the computation graphs, VGMEA supports flexible inner designs, e.g. adopts non-differentiable distributions or changes the number of iterations, that will not affect the original model architectures; (c) VGMEA is light-weighted and achieve their best results with only few parameters; (d) VGMEA supports high resolutions of the variable input sizes with relatively low computational complexity. We conduct extensive comparisons with baselines to demonstrate that different modules can be generically interpreted as specific instances of our proposed framework. Moreover, comprehensive experiments across multiple public datasets clearly demonstrate the performance of the derived methods. To summarize, the contribution of this chapter is the following: The remainder of the chapter is organized as follows. Section 3.2 introduces all related work. In Section 3.3, we illustrate the proposed framework and its interpretation, and in Section 3.4, we present our model in detail. In Section 3.5, we demonstrate the performance of the model, including comparison experiments, experimental evaluation, and analysis. Finally, the conclusion and future follow-up research are presented in Section 3.6.

3.2 Related Work

3.2.1 Variational Bayesian methods

Variational Bayes (VB), also called Variational Inference or Variational Approximation, is a family of graphical models for approximating intractable distributions arising in Bayesian inference. The node set typically consists of observed variables (i.e., data) and unobserved variables (i.e., parameters and latent variables), while the edge set represents various sorts of dependencies among all variables. VB can be seen as an extension of the expectation-maximization (EM) algorithm that upgrades maximum a posteriori (MAP) estimation to fully Bayesian estimation. Instead of performing point estimation on each parameter, VB computes an analytical approximation to the entire posterior distribution of unobserved variables. The approximated distribution, called the variational distribution, is usually a factorization of the true posterior distribution. Each factor can be restricted to a family of distributions with a simpler form at the cost of cutting off some edges. To increase approximation precision, extra variational

parameters are subsequently attached to factors, model the missing dependencies, and are absorbed into each corresponding factor [195]. VB optimizes the evidence lower bound (ELBO) to alternatively recognize the variational distribution and update parameters. Although VB is scalable to large data sets and applicable to inferring complex models with a large number of parameters, it has not been widely used in the deep learning community because a large number of trials is often required to find a suitable factorization. If the variational distribution has been determined, VB produces locally optimal but exact analytical solutions at a faster rate than sampling methods, e.g., Markov Chain Monte Carlo (MCMC) and Sequential Monte Carlo (SMC). Our VGMEA model introduces a set of extra variational parameters to support complicated regularization terms, e.g., total-variance, to force attention maps to be smooth and compact. Moreover, the proposed model adopts local variational methods [37] to ensure that the posterior distribution is the conjugate of the prior distribution. In other words, the Bayesian update of our approach can be safely reformulated as an incremental mini-batch update, which better aligns with deep neural networks and achieves fast and stable convergence. Besides, VB can derive a lower bound for the marginal likelihood of the observed data [2], which is typically used as a metric to select among models. The general idea is that a higher marginal likelihood of a model indicates a better fit to the data and, subsequently, a greater probability of generating the data [443].

3.2.2 Clustering Algorithms

Clustering methods aim to group a set of objects so that objects in the same cluster are more conceptually similar to each other than to those in other clusters. A feature vector represents each object to describe its task-specific attributes, while the conceptual similarity is quantified as a spatial distance between feature vectors by a given metric. Based on the distance function, all feature vectors are mapped onto a feature space, depicted in Figure 3.3b. Clustering methods are simple yet general algorithms that can be applied to arbitrary feature spaces to distinguish groups of concepts. According to different viewpoints, there are mainly two categories of clustering approaches: k-nearest neighbors (k-NN) and K-means-style clustering methods.

K-NN style clustering is a family of bottom-up algorithms with two hyperparameters. One hyperparameter controls the search scope, while the other determines the fusion thresholds. Given two parameters, the approach directly uses the original feature vectors and computes a center for each object among corresponding nearby

[2] sometimes called the evidence, i.e., the marginal probability of the data by performing an integral over unobserved variables

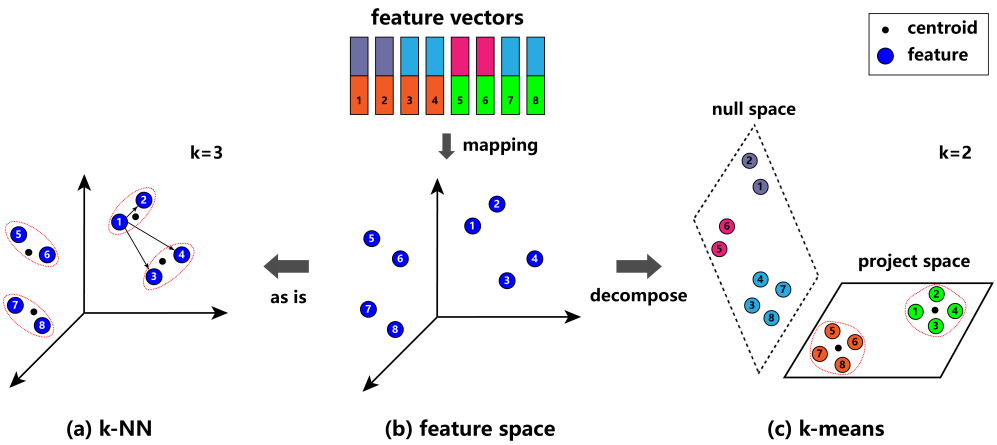


Figure 3.3: A visualization of clustering algorithms. There are 8 feature vectors, divided into 2 parts, each with an index. The lower part represents common information, and the upper part represents individual information. Different colors of features indicate greater distance in the feature space. Therefore, the first 4 feature vectors are from one class and the rest from the other. Moreover, every 2 feature vectors from one object. (a) k-NN clustering directly uses the original feature vectors and assigns each point to its 3 nearest neighbors. Through subtle differences between the individual information, 2 dots belong to one center; (b) All feature vectors are mapped as blue dots onto feature space; (c) K-means clustering divided each feature vector into two parts. The lower parts (i.e., common information) are projected onto an embedding space. In contrast, the upper parts are discarded into a null space. 2 global centers for 2 classes are obtained via significant differences between the common information.

objects, depicted in Figure 3.3a. In other words, k-NN style clustering highlights subtle differences among spatially close features and computes adaptive centers for each input feature. Notably, some centers can be very close to each other. When pairwise center distances are below a given fusing threshold, those centers will be regarded as duplicates and fused. Therefore, the effective number of centers depends on the input features.

In contrast, K-means style clustering is a family of top-down algorithms with a single hyperparameter: the number of centers. Given the number of centers, the method decomposes the original feature space into a null space and an embedding space. The embedding space contains common information, whilst the null space contains the neglected (also called noise). Subsequently, centers are computed within the embedding space, depicted in Figure 3.3c. Therefore, K-means clustering targets significant differences among the common parts and shares centers for all input features.

In summary, K-means style clustering can distill globally shared information, but k-NN style clustering can focus on locally subtle information. Although they are interchangeable in many application scenarios, we argue that K-means style clustering is more suitable for semantic segmentation, and K-NN style clustering for object

detection/instance segmentation. Indeed, K-means can increase the number of centers for object-level clustering, but it is inefficient and will fail when objects are dense [725]. Likewise, k-NN can expand its search scope to find common information, but its variable clusters might be under- or over-clustering of semantic concepts.

3.2.3 Attention Mechanism

Attention mechanisms, first as self-attention, originated from natural language processing (NLP) for sequential data (e.g., speech, language) [642], and then were generalized as a non-local operator [678] for spatial (e.g., image) and temporal-spatial data (e.g., video). Thanks to capturing *long-range* dependencies between pairwise positions, self-attention has achieved promising results on vision-related tasks, including image retrieval [74, 175], image classification [33, 601, 508, 285, 757, 453, 547], and visual tracking [871]. However, self-attention uses a single feature map as input and computes only long-range interactions within a single scale, thereby failing to capture broader dependencies across feature maps at different scales. Therefore, cross-attention was proposed by CrossViT [66], and includes self-attention as a specific instance. Cross-attention takes two feature maps from multiple scales as inputs and computes long-range dependencies across scales. This extension has further improved the performance on visual tasks, including image super-resolution [486, 296, 129], semantic segmentation [184, 287, 448], and instance segmentation [177, 109, 209]. However, these attention modules need to produce large attention maps, which, in turn, results in high computational complexity. The bottleneck lies in the fact that both the generation of the attention map and its use are computed across all positions.

There are many improvements towards addressing the issue. EncNet [806] defined k code-words and computed attention as Euclidean distances between input feature maps and k code-words. EMA [379] reformulated attention as a reconstruction problem and computed a more compact basis set via the expectation maximization algorithm. OCRNet [784] simplified attention to a pure clustering with given learned segmentation masks. Deformable attention [870] sampled at k learnable offsets, and computed correlations between the center and its k selected neighbors. Mask2former [105] defined k global queries, and computed attention as Cosine similarity between k queries and the input *key* feature maps. Max-DeepLab [654, 773] conducted attention as a hard assignment of k-means clustering. K-NN attention [663] adopted top k most similar positions. Our model is a probabilistic model that can interpret K-means attention modules as specific instances. Moreover, multi-head attention can be explained as

independent block matrix computation, and the mini-batch training strategy can work well with the Bayesian update in our model.

3.3 Variational Gaussian Mixture Embedded Attention

In this section, we introduce the variational Gaussian mixture embedded attention (VGMEA) to interpret the attention mechanism in a probabilistic way. A VGMEA is an unifying framework that can capture long-range information across scales, distill global concepts, and compute a residual to highlight semantically oriented information. Given multi-scale feature maps $X = \{X_s\}_{s=1}^{S}$ that extracts from an input image $I \in \mathbb{R}^{H \times W \times 3}$ by a backbone network, where $X_s \in \mathbb{R}^{C \times N_s}$ is a feature map at scale s with $N_s = H_s \times W_s$ spatial positions. We define the VGMEA as a mapping $F : X \to Y$, where $Y \in \mathbb{R}^{C \times N}$ are enhanced and semantically oriented feature maps distilled from semantic concepts.

VGMEA updates the following global information.

Semantic concept centroids:

$$\boldsymbol{\mu} = \{\boldsymbol{\mu}_k\}_{k=1}^{K} \in \mathbb{R}^{K \times C}$$

Semantic correlation matrix:

$$\mathbf{C} \in \mathbb{R}^{K \times K}$$

Scale-aware attention maps:

$$A = \{A_s \in \mathbb{R}^{N_s \times K}\}_{s=1}^{S}$$

VGMEA outputs enhanced semantic-oriented feature maps $Y \in \mathbb{R}^{C \times N}$:

$$Y_s = A_s \cdot \mathbf{C}^{-1} \cdot \boldsymbol{\mu} + X_s \tag{3.1}$$

where S, C, N, and K are the number of scales, channels, positions, and bases, respectively. Noted that the multi-scale attention maps represent scale-aware concept assignment.

3.3.1 Problem Formulation and Notation

Denote $\Theta = \{\Phi, m, \beta, W, \nu\}$ as a set of learnable parameters in the VGMEA module with an initial value $\Theta_0 = \{\Phi_0, m_0, \beta_0, W_0, \nu_0\}$. Besides, $(\boldsymbol{\mu}, \mathbf{C})$ is the distilled global concepts and $Z = [Z_1, Z_2, \ldots, Z_K]$ is a latent random field to estimate the multi-scale attention maps A for concept assignment. Specifically, each position of Z is a 1-of-K vector $Z_k \in \{0, 1\}$ and satisfies $\sum_{k=1}^{K} Z_k = 1$. Then, the probabilistic framework of the proposed ***variational Gaussian Mixture model with geometric regularization*** is defined as follows:

$$
\begin{aligned}
P_\Theta(X, Z, \boldsymbol{\mu}, \mathbf{C}) &= P(X|Z, \boldsymbol{\mu}, \mathbf{C}) \cdot P_\Theta(Z) \cdot P_\Theta(\boldsymbol{\mu}, \mathbf{C}) \\
P(X|Z, \boldsymbol{\mu}, \mathbf{C}) &= \prod_{n=1}^{N} \prod_{k=1}^{K} \mathcal{N}(X_n|\boldsymbol{\mu}_k, \mathbf{C}_k^{-1})^{Z_{n,k}} \\
P_\Theta(Z) &= F(\Phi) \cdot \prod_{n=1}^{N} \prod_{k=1}^{K} \delta(\phi_{n,k})^{Z_{n,k}} \cdot \delta(-\phi_{n,k})^{1-Z_{n,k}} \\
P_\Theta(\boldsymbol{\mu}, \mathbf{C}) &= \prod_{k=1}^{K} \mathcal{N}\left(\boldsymbol{\mu}_k|m, (\beta \cdot \mathbf{C}_k)^{-1}\right) \mathcal{W}(\mathbf{C}_k|W, \nu)
\end{aligned}
\tag{3.2}
$$

where $\mathcal{N}(\cdot)$ is a Gaussian distribution, $\mathcal{W}(\cdot)$ is a Wishart distribution. In addition, $P_\Theta(Z)$ is a Categorical distribution with a normalization term $F(\Phi)$ and $\delta(\cdot)$ is a sigmoid function. Based on auxiliary Φ, we can regulate the random field Z as a smooth manifold and enforce $P(X|Z, \boldsymbol{\mu}, \mathbf{C})$ generating robust label affiliation A which has continuity within semantic regions and sharp boundaries between.

3.3.2 Variational Bayesian Inference with Geometric Regularization

Set $Q_\Theta(Z, \boldsymbol{\mu}, \mathbf{C}) = Q_\Theta(Z) \cdot Q_\Theta(\boldsymbol{\mu}, \mathbf{C})$ as a lower bound of posterior distribution. We define $T \in \{Z \times \boldsymbol{\mu} \times \mathbf{C}\}$ as an infinitesimal variable in a Descartes product space. For simplicity, we represents $\tilde{T} = T \setminus S = \{\boldsymbol{\mu} \times \mathbf{C}\}$.

Variational Inference to Approximate Posterior Distributions

Denote the evidence lower bound (ELBO) as **K**ullback–**L**eibler divergence with an

additive geometric regularization to measure the gap:

$$\mathrm{KL}(Q||P) = \int_T Q_\Theta(Z) Q_\Theta(\boldsymbol{\mu}, \mathbf{C}) \ln \frac{P_\Theta(X, Z, \boldsymbol{\mu}, \mathbf{C})}{Q_\Theta(Z, \boldsymbol{\mu}, \mathbf{C})} \, \mathrm{d}\, T + \lambda \, \mathrm{Reg}(\Phi) \tag{3.3}$$

Noted that it is safe to ignore the geometric term when approximating the posterior functionals because $\mathrm{Reg}(\Phi)$ remains a constant during variational inference. Using the definition of expectation and rearranging the order of integration, we obtain the following two equations.

$$\begin{aligned} \mathrm{KL}(Q||P) &= \int_Z Q_\Theta(Z) \cdot \int_{\tilde{T}} Q_\Theta(\boldsymbol{\mu}, \mathbf{C}) \ln \frac{P_\Theta(X, Z, \boldsymbol{\mu}, \mathbf{C})}{Q_\Theta(Z, \boldsymbol{\mu}, \mathbf{C})} \, \mathrm{d}\, \tilde{T} \, \mathrm{d}\, Z \\ &\propto \int_Z Q_\Theta(Z) \cdot \ln \frac{P_\Theta(Z) \cdot \exp\{\mathbb{E}_{\tilde{T}}[\ln P(X|Z, \boldsymbol{\mu}, \mathbf{C})]\}}{Q_\Theta(Z)} \, \mathrm{d}\, Z \end{aligned} \tag{3.4}$$

where exp is an exponential function, and

$$\begin{aligned} \mathrm{KL}(Q||P) &= \int_{\tilde{T}} Q_\Theta(\boldsymbol{\mu}, \mathbf{C}) \cdot \int_Z Q_\Theta(Z) \ln \frac{P_\Theta(X, Z, \boldsymbol{\mu}, \mathbf{C})}{Q_\Theta(Z, \boldsymbol{\mu}, \mathbf{C})} \, \mathrm{d}\, Z \, \mathrm{d}\, \tilde{T} \\ &\propto \int_{\tilde{T}} Q_\Theta(\boldsymbol{\mu}, \mathbf{C}) \cdot \ln \frac{P_\Theta(\boldsymbol{\mu}, \mathbf{C}) \cdot \exp\{\mathbb{E}_Z[\ln P(X|Z, \boldsymbol{\mu}, \mathbf{C})]\}}{Q_\Theta(\boldsymbol{\mu}, \mathbf{C})} \, \mathrm{d}\, \tilde{T} \end{aligned} \tag{3.5}$$

In order to minimize the gap, we have to solve the following equations:

$$\begin{aligned} \frac{P_\Theta(Z) \cdot \exp\{\mathbb{E}_{\tilde{T}}[\ln P(X|Z, \boldsymbol{\mu}, \mathbf{C})]\}}{Q_\Theta(Z)} &= 1 \\ \frac{P_\Theta(\boldsymbol{\mu}, \mathbf{C}) \cdot \exp\{\mathbb{E}_Z[\ln P(X|Z, \boldsymbol{\mu}, \mathbf{C})]\}}{Q_\Theta(\boldsymbol{\mu}, \mathbf{C})} &= 1 \end{aligned} \tag{3.6}$$

Therefore, we obtain optimal functionals, where Con is a normalization constant for probability functions.

$$Q_\Theta(Z) = P_\Theta(Z) \cdot \exp\{\mathbb{E}_{\tilde{T}}[\ln P(X|Z, \boldsymbol{\mu}, \mathbf{C})] \cdot \mathrm{Con} \tag{3.7}$$

$$Q_\Theta(\boldsymbol{\mu}, \mathbf{C}) = P_\Theta(\boldsymbol{\mu}, \mathbf{C}) \cdot \exp\{\mathbb{E}_Z[\ln P(X|Z, \boldsymbol{\mu}, \mathbf{C})] \cdot \mathrm{Con} \tag{3.8}$$

Hereafter, we insert the previous probability functions into Eq.3.7 and Eq.3.8. Then we recognize the approximate posterior function $Q_\Theta(Z)$ as a ***Categorical*** distribution:

$$
\begin{aligned}
Q_\Theta(Z) &= \prod_{n=1}^{N}\prod_{k=1}^{K} \underbrace{e^{\left\{\phi_{n,k}+\mathbb{E}_{\tilde{T}}\left[\ln \mathcal{N}(X_n|\boldsymbol{\mu}_k, C_k^{-1})\right]\right\}}}_{\rho_{n,k}}{}^{Z_{n,k}} \cdot \mathrm{Con} \\
&= \prod_{n=1}^{N}\prod_{k=1}^{K} \rho_{n,k}^{Z_{n,k}} \cdot \mathrm{Con} \\
&= \prod_{n=1}^{N}\prod_{k=1}^{K} \underbrace{\left(\frac{\rho_{n,k}}{\sum_{k=1}^{K}\rho_{n,k}}\right)}_{r_{n,k}}^{Z_{n,k}} \\
&= \prod_{n=1}^{N}\prod_{k=1}^{K} r_{n,k}^{Z_{n,k}}
\end{aligned}
\tag{3.9}
$$

Hence, we have the standard result

$$
\mathbb{E}_Z[Z_{n,k}] = r_{n,k} \tag{3.10}
$$

Noted that each quantity $r_{n,k}$ is a part of the softmax functions and represents a soft affiliation, which plays the role of responsibilities. Note that the optimal solution for $Q_\Theta(Z)$ depends on moments evaluated with respect to the distributions of other variables. Therefore, the variational update equations are coupled and must be solved iteratively. At this point, we shall find it convenient to define three statistics of the observed data set evaluated from the responsibilities, given by

$$
\begin{aligned}
N_k &= \sum_{n=1}^{N} r_{n,k} \\
\bar{X}_k &= \frac{1}{N_k}\sum_{n=1}^{N} r_{n,k} X_n \\
\mathrm{S}_k &= \frac{1}{N_k}\sum_{n=1}^{N} r_{n,k}(X_n - \bar{X}_k)(X_n - \bar{X}_k)^T
\end{aligned}
\tag{3.11}
$$

Moreover, the variational posterior distribution $Q_\Theta(\boldsymbol{\mu}, \mathbf{C})$ is recognized as a ***Gaussian-Wishart*** distribution:

$$
\begin{aligned}
&Q_\Theta(\boldsymbol{\mu}, \mathbf{C}) \\
&\quad \propto \prod_{k=1}^{K} \mathcal{N}\left(\boldsymbol{\mu}_k | m, (\beta C_k)^{-1}\right) \cdot \mathcal{W}(C_k | W, \nu) \cdot \prod_{n=1}^{N} e^{\left\{\mathbb{E}_Z\left[Z_{n,k} \cdot \ln \mathcal{N}(X_n | \boldsymbol{\mu}_k, C_k^{-1})\right]\right\}} \\
&\quad \propto \prod_{k=1}^{K} \mathcal{N}\left(\boldsymbol{\mu}_k | m, (\beta C_k)^{-1}\right) \cdot \mathcal{W}(C_k | W, \nu) \cdot \prod_{n=1}^{N} e^{r_{n,k}} \cdot \mathcal{N}(X_n | \boldsymbol{\mu}_k, C_k^{-1}) \\
&\quad = \prod_{k=1}^{K} \mathcal{N}(\boldsymbol{\mu}_k | m_k, (\beta_k C_k)^{-1}) \cdot \mathcal{W}(C_k | W_k, \nu_k)
\end{aligned}
\tag{3.12}
$$

where m_k, β_k, W_k and ν_k are parameters for the posterior distribution. Based on the momentum statistics, the optimal parameters can be updated as

$$
\begin{aligned}
&\beta_k = \beta_0 + N_k \\
&m_k = \frac{\beta_0 m_0 + N_k \bar{X}_k}{\beta_0 + N_k} \\
&{W_k}^{-1} = W_0^{-1} + N_k \mathrm{S}_k + \frac{\beta_0 N_k}{\beta_0 + N_k} (\bar{X}_k - m_0)(\bar{X}_k - m_0)^T \\
&\nu_k = \nu_0 + N_k
\end{aligned}
\tag{3.13}
$$

Geometric Regularization in Concept Space

In addition, our variational probabilistic model is flexible to integrate geometric regularization terms Φ for the regularity of the responsibility $r_{n,k}$. For simplicity, we integrate total variance (TV) as an example. Denote the likelihood function as $L(\Theta, \Theta^\star)$. Under the expectation conception, all hidden random variables become constants. Therefore, the likelihood function only contains geometric regularization parameters for optimization.

$$
\begin{aligned}
L(\Theta, \Theta^\star) &= \int_T Q_{\Theta^\star}(Z, \boldsymbol{\mu}, \mathbf{C}) \ln \frac{P_\Theta(X, Z, \boldsymbol{\mu}, \mathbf{C})}{Q_{\Theta^\star}(Z, \boldsymbol{\mu}, \mathbf{C})} \, \mathrm{d}\, T + \lambda \, \mathrm{Reg}(\Phi) \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} r_{n,k} \cdot \phi_{n,k} + \ln \delta(-\phi_{n,k}) + \lambda \, \mathrm{Reg}(\phi_{n,k}) + \mathrm{Con.} \\
&= \sum_{n=1}^{N} \sum_{k=1}^{K} r_{n,k} \cdot \phi_{n,k} + \ln \delta(-\phi_{n,k}) + \lambda \cdot |\nabla \delta(\phi_{n,k})| + \mathrm{Con.}
\end{aligned}
\tag{3.14}
$$

Where Con. contains other irrelevant model parameters. Then we compute the derivative and set it to zero:

$$\begin{aligned}
\frac{\partial L(\Theta, \Theta^\star)}{\partial \phi_{n,k}} &= \frac{\mathrm{d}}{\mathrm{d}\,\phi_{n,k}} \left[r_{n,k} \cdot \phi_{n,k} + \ln \delta(-\phi_{n,k}) + \lambda \cdot |\nabla \delta(\phi_{n,k})| \right] \\
&= r_{n,k} - \delta(\phi_{n,k}) + \lambda \cdot \delta'(\phi_{n,k}) \cdot \underbrace{\mathrm{div}(\frac{\nabla \phi_{n,k}}{|\nabla \phi_{n,k}|})}_{\kappa_{n,k}} \\
&= r_{n,k} - \delta(\phi_{n,k}) + \lambda \cdot \delta'(\phi_{n,k}) \cdot \kappa_{n,k} \\
&= 0
\end{aligned} \tag{3.15}$$

It is a difficult differential equation to solve. However, we can follow the routine of the level-set method that sets $\frac{\partial L(\Theta,\Theta^\star)}{\partial \phi_{n,k}} = -\frac{\mathrm{d}\,\phi_{n,k}}{dt}$. Then we have a momentum-based update equation:

$$\begin{aligned}
\frac{\mathrm{d}\,\phi_{n,k}}{dt} &= \delta(\phi_{n,k}) - r_{n,k} - \lambda \cdot \delta'(\phi_{n,k}) \kappa_{n,k} \\
\rightarrow \phi_{n,k}^{t+1} &= \phi_{n,k}^{t} + \nabla t \left[\underbrace{\delta(\phi_{n,k})}_{\text{sigmoid}} - \overbrace{r_{n,k}}^{\text{softmax}} - \lambda \underbrace{\delta'(\phi_{n,k}) \kappa_{n,k}}_{\text{smoothness}} \right]
\end{aligned} \tag{3.16}$$

where $\nabla t \geq 0$ is a scaling factor to update the manifold surface and λ is a weight for smoothness regularization. Noted that the geometric regularization introduces smoothness into the softmax responsibility function and generates light-tailed distributions. It is beneficial to generate piece-wise smoothing label affiliation maps for $r_{n,k}$, i.e., consistent assignment within each semantic region with sharp interval boundaries.

3.3.3 Cross-Attention as Concept Distillation

Modern vision transformers implement a cross-attention module as a series of blocks. Each block adopts a residual structure and updates the query once. Hence, the entire cascade cross-attention module performs a unified formulation for iterative concept refinement. Given query features $\mathbf{Q} \in \mathbb{R}^{N_q \times C}$, key features $\mathbf{K} \in \mathbb{R}^{N_k \times C}$, and value

features $\mathbf{V} \in \mathbb{R}^{N_k \times C}$, the iterative residual formulation can be reformulate as:

$$\begin{aligned} \nabla \mathbf{Q}^{(l)} &= \text{softmax}\left(\frac{\mathbf{Q}^{(l)}\mathbf{K}^\top}{\sqrt{d_k}}\right) \cdot \mathbf{V} \\ \mathbf{Q}^{(l+1)} &= \mathbf{Q}^{(l)} + \nabla \mathbf{Q}^{(l)} \end{aligned} \tag{3.17}$$

where l represents the block index. This operation can be interpreted as a mean-shift process that sequentially computes the query shifts $\nabla \mathbf{Q}$ and updates the query using a momentum update with even weights 1.

The proposed VGMEA generalizes the mean-shift process as a joint probabilistic model and interprets it through EM algorithms:

$$\begin{aligned} \textbf{E-step (Assignment):} \quad & r_{n,k}^{(l)} = \mathbb{E}_Z[Z_{n,k}] \\ \textbf{M-step (Update):} \quad & \boldsymbol{\mu}_k^{(l+1)} = \boldsymbol{\mu}_k^{(l)} + \eta \bar{X}_k \\ & \mathbf{C}_k^{(l+1)} = \mathbf{C}_k^{(l)} + \eta \bar{S}_k \end{aligned} \tag{3.18}$$

where $\mathbb{E}_Z[Z_{n,k}]$ is the expectation of the approximated posterior distribution. η is the momentum update factor. $\bar{X}_k$ and $\bar{S}_k$ are sampled mean and correlation estimated from responsibility $r_{n,k}$, respectively.

Method	Assignment	Update	Correspondence
Non-local*[678]	$r_{i,j} = \Upsilon\left(e^{\boldsymbol{x}_i^T \boldsymbol{q}_j}\right)$	$\boldsymbol{\mu}_j = W_z \cdot \bar{X}_j + \boldsymbol{q}_j$, $\mathbf{C}_j \equiv I$	$\eta = W_z$, $\bar{X}_j = \sum r_{i,j} \cdot g(\boldsymbol{x}_i)$, $\phi_i \equiv 0$
EncNet[806]	$r_{i,j} = \Upsilon\left(e^{-s_k \lVert \boldsymbol{d}_{i,j} \rVert^2}\right)$	$\boldsymbol{\mu_k} = h(r_{i,j} \cdot \boldsymbol{d}_{i,j})$, $\mathbf{C}_j \equiv I$	$\eta = 0$, $\bar{X}_j \equiv 0$, $\phi_i \equiv 0$
EMANet[379]	$r_{i,j} = \Upsilon\left(e^{-(x_i - \boldsymbol{q}_j)^T \mathbf{C}_j^{-1}(x_i - \boldsymbol{q}_j)}\right)$	$\boldsymbol{\mu}_j = \bar{X}_j$, $\mathbf{C}_j = \boldsymbol{S}_j$	$\bar{X}_j = N_j^{-1} \sum r_{i,j} \cdot \boldsymbol{x}_i$, $\phi_i \equiv 0$ $\boldsymbol{S}_j = N_j^{-1} \sum r_{i,j} \cdot D^2(\boldsymbol{d}_{i,j})$
OCRNet[784]	$r_{i,j} = \boldsymbol{m}_{i,j}$	$\boldsymbol{\mu}_j = \bar{X}_j$, $\mathbf{C}_j \equiv I$	$\bar{X}_j = \sum \boldsymbol{m}_{i,j} \boldsymbol{x}_i$, $\phi_i \equiv 0$
kMax-DeepLab[773]	$r_{i,j} = \arg\max(\mathbf{q}_j \mathbf{k}_i^T)$	$\boldsymbol{\mu}_j = \boldsymbol{q}_j + \bar{X}_j$, $\mathbf{C}_j \equiv I$	$\eta = 1$, $\bar{X}_j = \sum r_{i,j} \cdot \boldsymbol{v}_i$, $\phi_i \equiv 0$
MaskFormer[106]	$r_{i,j} = \Upsilon(q_j k_i^T)$	$\boldsymbol{\mu}_j = \boldsymbol{q_j} + \bar{X}_j$, $\mathbf{C}_j \equiv I$	$\eta = 1$, $\bar{X}_j = \sum r_{i,j} \cdot \boldsymbol{v}_i$, $\phi_i \equiv 0$
Mask2Former[105]	$r_{i,j} = \Upsilon(\mathbf{q}_k \mathbf{k}_i^T + \mathcal{M}_i)$	$\boldsymbol{\mu}_j = \boldsymbol{q_j} + \bar{X}_j$, $\mathbf{C}_j \equiv I$	$\eta = 1$, $\bar{X}_j = \sum r_{i,j} \cdot \boldsymbol{v}_i$, $\phi_i = \mathcal{M}_i$

Table 3.1: Special Interpret SOTA models as VGMEA Instances. (1) Non-local[678] is implemented as the cross-attention and $g(\cdot)$ denotes transformation function. (2) $\Upsilon(\cdot) = \text{softmax}(\cdot)$, $\text{m}_{i,k}$ denotes a semantic segmentation prediction from a sub-network. (3) $h(\cdot)$ denotes Batch Normalization with ReLU activation and $\boldsymbol{d}_{i,k} = \boldsymbol{x}_i - \boldsymbol{q}_k$. (4) $\kappa(\boldsymbol{x}, \boldsymbol{y}) = f_1(\boldsymbol{x})^T f_2(\boldsymbol{y})$ where $f_1(\cdot)$ and $f_2(\cdot)$ are two transformation functions. (5) $\mathcal{M}_i$ is the binarized output of the resized mask prediction, $D^2(\boldsymbol{x}) = \boldsymbol{x} \cdot \boldsymbol{x}^T$.

To demonstrate generalization, we analyze a few state-of-the-art models as special

cases of VGMEA and organize them in Table 3.1. We observe that all methods set the correlation matrix $\mathbf{C}$ to the identity matrix, except EMANet, suggesting that semantic centroids are orthonormal. Furthermore, none of the methods can support geometrical constraints to further sharpen cross-class decision boundaries, except Mask2Former. More specifically, Non-local[678] is a generalized K-means clustering algorithm that allows transformations between different feature representations. However, it cannot be stacked to update semantic concepts $\boldsymbol{\mu}$ due to its high computational complexity. In addition, EncNet[806] defined a global semantic codebook and computed cross-attention by averaging the residuals between the input features and the codebook. The averaged residuals are interpreted as context encoding to guide semantic segmentation, suppressing absent classes. Nevertheless, EncNet cannot extract sample statistics $\bar{X}$ from the input feature maps and therefore fails to adapt the context encoding to the input. In contrast, EMANet[379] only considers sample statistics and sample correlation, resulting in vulnerability to the batch size and data noise. In contrast, OCRNet [784] is an amortized K-means clustering algorithm that relies heavily on predicted semantic segmentation masks to compute the statistics $\bar{X}$. It used $\bar{X}$ as dynamic queries to compute cross-attention. However, its bottleneck is the quality of mask prediction and the batch size. kMax-DeepLab[773] follows the conventional K-means clustering algorithms and adopts hard assignment to compute sample statistics $\bar{X}$. MaskFormer[106] is a simplified version that does not use geometrical constraints to ensure cross-class decision boundaries. Likewise, Mask2Former [106] adopts amortized constraints represented as binarized mask predictions. In summary, our VGMEA is a generalized probabilistic framework that can accommodate a wide range of cross-attention mechanisms.

3.4 Implementation Details

3.4.1 Online Bayesian Updates for Batch Processing

Deep learning pipelines process data in mini-batches. However, traditional EM requires full-dataset iterations, making it incompatible with stochastic optimization. Therefore, we reformulate VGMEA with the Bayesian formula to support incremental/iterative updates.

Given a dataset of M independent batches $X = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_M\}$ and any mini-

batch $\mathcal{B}_i$, the probabilistic model is equal to a chain of conditional distributions:

$$\begin{aligned}
P_\Theta(X, Z, \boldsymbol{\mu}, \mathbf{C}) &= P_\Theta(X \setminus \mathcal{B}_i, \mathcal{B}_i, Z, \boldsymbol{\mu}, \mathbf{C}) \\
&= P_\Theta(\mathcal{B}_i, Z, \boldsymbol{\mu}, \mathbf{C} | X \setminus \mathcal{B}_i) \cdot P(X \setminus \mathcal{B}_i) \\
&= P_\Theta(\mathcal{B}_i | Z, \boldsymbol{\mu}, \mathbf{C}, X \setminus \mathcal{B}_i) \cdot P_\Theta(Z, \boldsymbol{\mu}, \mathbf{C} | X \setminus \mathcal{B}_i) \cdot P(X \setminus \mathcal{B}_i) \\
&\propto P_\Theta(\mathcal{B}_i | Z, \boldsymbol{\mu}, \mathbf{C}, X \setminus \mathcal{B}_i) \cdot P_\Theta(Z, \boldsymbol{\mu}, \mathbf{C} | X \setminus \mathcal{B}_i) \\
&\approx P_\Theta(\mathcal{B}_i | Z, \boldsymbol{\mu}, \mathbf{C}, X \setminus \mathcal{B}_i) \cdot P_\Theta(Z | X \setminus \mathcal{B}_i) \cdot P_\Theta(\boldsymbol{\mu}, \mathbf{C} | X \setminus \mathcal{B}_i) \\
&= P_\Theta(\mathcal{B}_i | Z, \boldsymbol{\mu}, \mathbf{C}, X \setminus \mathcal{B}_i) \cdot Q_\Theta^{(t-1)}(Z) \cdot Q_\Theta^{(t-1)}(\boldsymbol{\mu}, \mathbf{C})
\end{aligned} \tag{3.19}$$

where $Q_\Theta^{(t-1)}(Z)$ and $Q_\Theta^{(t-1)}(\boldsymbol{\mu}, \mathbf{C})$ are the approximated variational posterior distributions based on an arbitrary subset of the dataset. Thanks to the conjugate priors $Q_\Theta^{(t-1)}(Z)$ and $Q_\Theta^{(t-1)}(\boldsymbol{\mu}, \mathbf{C})$, the generative distribution can be approximated as a VGMEA at any bipartite dataset. Formally, the VGMEA can adapt the EM algorithm into an online update scheme to compute the posterior distribution:

$$\begin{aligned}
P_\Theta(X \setminus \mathcal{B}_i, Z, \boldsymbol{\mu}, \mathbf{C} | \mathcal{B}_i) &= \frac{P_\Theta(X \setminus \mathcal{B}_i, \mathcal{B}_i, Z, \boldsymbol{\mu}, \mathbf{C})}{P(\mathcal{B}_i)} \\
&\propto P_\Theta(\mathcal{B}_i | Z, \boldsymbol{\mu}, \mathbf{C}, X \setminus \mathcal{B}_i) \cdot Q_\Theta^{(t-1)}(Z) \cdot Q_\Theta^{(t-1)}(\boldsymbol{\mu}, \mathbf{C}) \\
&\approx Q_\Theta^{(t)}(Z) \cdot Q_\Theta^{(t)}(\boldsymbol{\mu}, \mathbf{C})
\end{aligned} \tag{3.20}$$

with conjugate priors

$$\begin{aligned}
&P_\Theta(\mathcal{B}_i | Z, \boldsymbol{\mu}, \mathbf{C}, X \setminus \mathcal{B}_i) \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C}) \\
&Q_\Theta^{(t-1)}(\mu_k, C_k) \sim \mathcal{N}\left(m_k^{(t)}, (\beta_k^{(t)} C_k)^{-1}\right) \cdot \mathcal{W}\left(W_k^{(t)}, \nu_k^{(t)}\right) \\
&Q_\Theta^{(t-1)}(Z_{n,k}) \sim \mathrm{Cat}(\phi_{n,k})
\end{aligned} \tag{3.21}$$

Therefore, we can formulate the standard incremental update rules for the VGMEA.

Let $(N_k^{(\mathcal{B})}, \bar{X}_k^{(\mathcal{B})}, S_k^{(\mathcal{B})})$ be sufficient statistics at mini-batch $\mathcal{B}$:

$$\begin{aligned} N_k^{(\mathcal{B})} &= \sum_{i \in \mathcal{B}} r_{i,k} \\ \bar{X}_k^{(\mathcal{B})} &= \frac{1}{N_k^{(\mathcal{B})}} \sum_{i \in \mathcal{B}} r_{i,k} X_i \\ S_k^{(\mathcal{B})} &= \frac{1}{N_k^{(\mathcal{B})}} \sum_{i \in \mathcal{B}} r_{i,k} \left(X_i - \bar{X}_k^{(\mathcal{B})} \right) \left(X_i - \bar{X}_k^{(\mathcal{B})} \right)^T \end{aligned} \tag{3.22}$$

Therefore, the model parameters $\mathbf{\Theta} = \{\mathbf{m}_k, \beta_k, \mathbf{W}_k, \nu_k\}$ can be updated as:

$$\begin{aligned} \beta_k^{(t)} &= \beta_k^{(t-1)} + N_k^{(\mathcal{B})} \\ m_k^{(t)} &= \frac{\beta_k^{(t-1)} m_k^{(t-1)} + N_k^{(\mathcal{B})} \bar{X}_k^{(\mathcal{B})}}{\beta_k^{(t-1)} + N_k^{(\mathcal{B})}} \\ ({W_k}^{(t)})^{-1} &= (W_k^{(t-1)})^{-1} + N_k^{(\mathcal{B})} \mathrm{S}_k^{(\mathcal{B})} + \frac{\beta_k^{(t-1)} N_k^{(\mathcal{B})}}{\beta_k^{(t-1)} + N_k^{(\mathcal{B})}} \epsilon_{\mathcal{B}}^{(t-1)} \\ \nu_k^{(t)} &= \nu_k^{(t-1)} + N_k^{(\mathcal{B})} \end{aligned} \tag{3.23}$$

where $\epsilon_{\mathcal{B}}^{(t-1)} = \left(\bar{X}_k^{(\mathcal{B})} - m_k^{(t-1)} \right) \left(\bar{X}_k^{(\mathcal{B})} - m_k^{(t-1)} \right)^T$. We have reformulated the EM algorithm to support mini-batch style based on the variational inference with Bayesian update formula. It implies the existence of a firm mathematical foundation for adapting large probabilistic models to deep learning models without losing generality.

3.4.2 Adaptions to Deep Learning

For efficiency and stability, we have adapted VGMEA for deep learning models.

Streaming update with Smoothing

To mitigate batch-size effects, we employ streaming averaging to update the sufficient

statistics:

$$\begin{aligned} N_k^{(t+1)} &= N_k^{(t)} + N_k^{(\mathcal{B})} \\ \bar{X}_k^{(t+1)} &= \alpha \bar{X}_k^{(t)} + (1-\alpha)\bar{X}_k^{(\mathcal{B})} \\ S_k^{(t+1)} &= \alpha S_k^{(t)} + (1-\alpha) S_k^{(\mathcal{B})} + O(\alpha^2)\epsilon_S^{(t)} \end{aligned} \tag{3.24}$$

where $\alpha = \frac{N_k^{(t)}}{N_k^{(t)}+N_k^{(\mathcal{B})}}$, $O(\alpha^2) = \frac{N_k^{(t)} N_k^{(\mathcal{B})}}{\left(N_k^{(t)}+N_k^{(\mathcal{B})}\right)^2}$ and $\epsilon_S^{(t)} = \left(\bar{X}_k^{(t)} - \bar{X}_k^{(\mathcal{B})}\right)\left(\bar{X}_k^{(t)} - \bar{X}_k^{(\mathcal{B})}\right)^T$. Noted that our streaming method can always achieve stable states with arbitrary mini-batch sizes, since $(1-\alpha) \to 0$ and $O(\alpha^2) \to 0$ when $N_k^{(t+1)}$ accumulate to large numbers with training process.

In addition, we introduce an extra momentum smoothing parameter $\gamma = 0.99$ to stabilize the early-stage training for convergence acceleration:

$$\begin{aligned} N_k^{(t+1)} &= \gamma N_k^{(t+1)} + (1-\gamma) N_k^{(t)} \\ \bar{X}_k^{(t+1)} &= \gamma \bar{X}_k^{(t+1)} + (1-\gamma)\bar{X}_k^{(t)} \\ S_k^{(t+1)} &= \gamma S_k^{(t+1)} + (1-\gamma) S_k^{(t)} \end{aligned} \tag{3.25}$$

Orthogonality Constraint

From the perspective of semantic segmentation, enforcing orthogonality among concept centroids is well-motivated both theoretically and practically. The core justification stems from the fundamental requirement that distinct semantic concepts should be linearly independent in the feature space. The precondition means that the feature representation of one class should not be representable as a linear combination of the features of other irrelevant classes. Linear independence is a prerequisite for a model to make unambiguous and confident predictions.

Orthogonality is a stronger and more desirable condition than mere linear independence. By enforcing $\boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j \approx 0$ for $i \neq j$, we ensure that the learned semantic concepts are maximally separable and reside in independent ranges of the feature space. Orthogonality directly mitigates feature entanglement, a common issue where models struggle to distinguish between visually similar classes. The orthogonality constraint acts as a powerful regularizer, guiding the learning process towards a feature space where the decision boundaries between classes are sharp and well-defined.

Computational advantages primarily drive the transition from encouraging lin-

ear independence to enforcing strict orthogonality. From a probabilistic standpoint, orthogonality allows us to make a critical simplification: it justifies the assumption of a diagonal precision matrix $\mathbf{C}_k^{-1}$. When concept centroids are orthogonal, the covariances between different feature dimensions are minimized. Consequently, the full Wishart distribution $\mathcal{W}(\mathbf{W}_k, \nu_k)$, which governs the prior over the full precision matrix, elegantly factorizes into a product of independent Gamma distributions for each diagonal element λ_{km}:

$$P(\mathbf{C}_k) = \mathcal{W}(\mathbf{C}_k^{-1}|W, \nu) = \prod_{m=1}^{K} \text{Gamma}(\lambda_{km}|\alpha_{km}, \beta_{km}) \tag{3.26}$$

This factorization improves computational efficiency in online Bayesian update for posterior inference by replacing Gaussian-Wishart distributions with Gaussian-Gamma distributions. The equivalence benefits the matrix inversion calculation for $\mathbf{W}_k^{-1}$ and $\mathbf{C}_k^{-1}$ in the mini-batch EM algorithm. Compared to updating two full inverse matrices, we can apply the reciprocal function only along all diagonal elements, reducing the complexity from $O(K^3)$ to $O(K)$. In practice, we enforce concept orthogonality via:

$$L_{\text{orth}} = \sum_{i \neq j} |\mu_i^T \mu_j| + \sum_k \|C_k - \text{diag}(C_k)\|_F^2 \tag{3.27}$$

where diag$(\cdot)$ is a diagonal matrix. The loss function ensures that semantic concepts are disentangled and that covariance matrices remain diagonal, thereby reducing computational complexity. In summary, orthogonality is not merely a mathematical convenience but also a structurally sound inductive bias for semantic segmentation. It ensures the learning of distinct, disentangled semantic concepts while enabling a highly efficient and scalable implementation of the probabilistic model by leveraging the computational simplifications afforded by diagonal covariance matrices.

Amortized Geometric Constraints

The geometric constraints can be a computational bottleneck in the VGMEA model, since iterative level-set Φ calculations typically require more than 15 iterations to converge from scratch for each new input, increasing training time by 50%. Therefore, we adopt amortized variational inference (AVI) for geometric constraints, reaching a balance between accuracy and speed. AVI amortizes the cost by finding an input-dependent initialization rather than solving from $\Phi = \mathbf{0}$. We train a single, lightweight neural network (the ϕ-predictor) to directly predict a coarse $\hat{\Phi}$ from each input feature

as a good start state. Only if the predicted $\hat{\Phi}$ is very close to the optimum, a short, subsequent refinement loop (e.g., 3 iterations instead of 15) is then sufficient to converge. The strategy narrows the "amortization gap" – the difference between the amortized and iterative solutions. The predictor network can be extremely simple – often just several convolutional layers. The setting requires a minimal number of parameters and FLOPs while yielding substantial computational savings by drastically reducing the number of refinement iterations. The forward pass of the predictor is computationally cheaper than a long optimization iteration.

In conclusion, from an engineering standpoint, AVI can serve as a learning-based approximation, a necessary architectural decision for deploying probabilistic models with complex constraints in practical computer vision systems. It successfully transforms an intractable, iterative inference problem into a tractable, feed-forward prediction problem. The minimal sacrifice in theoretical purity and a negligible drop in final performance are overwhelmingly justified by the orders-of-magnitude improvement in inference speed, predictability, and scalability. It makes a powerful but computationally intensive model like VGMEA feasible for improving attention maps, ensuring spatial coherence and alignment with object boundaries.

3.5 Experiment

3.5.1 Dataset and Metrics

Dataset: We conduct comprehensive evaluations on four challenging semantic segmentation benchmarks to assess the generality and robustness of our proposed method. PASCAL Context[484] is a detailed scene understanding dataset built upon PASCAL VOC 2010[167]. It provides pixel-wise semantic labels for the same images, extending beyond the core object classes. The dataset contains over 400 categories, including both "thing" (countable) and "stuff" (amorphous) classes, but we follow the convention of using only 59 classes for semantic segmentation. With 10,103 annotated images, it offers a more comprehensive understanding of an entire scene's context for benchmarking. Cityscapes [123] is a large-scale dataset focused on urban street scenes. It contains high-resolution images (2048×1024) with annotations across 19 semantic categories relevant for autonomous driving. The dataset provides 5000 fine-grained annotations (2,975 images for training, 500 for validation, and 1,525 for testing) and 20,000 extra coarse-grained annotations for training. Its emphasis on precise boundary delineation

makes it ideal for evaluating geometric accuracy. MS-COCO[415] is a large-scale object detection, segmentation, and captioning dataset. For semantic segmentation, we use the train2017, val2017, and test2017 splits, which contain 118,287, 5,000, and 40,670 images, respectively. The dataset features complex daily scenes with 80 object categories. Its large scale and dense, cluttered scenes make it a rigorous test for a model's robustness and generalization capability. ADE20K[850] is one of the most challenging benchmarks for semantic segmentation, containing 150 finely annotated categories including both objects and stuff. The dataset is split into 20,210 images for training, 2,000 for validation, and 3,352 for testing. Its large number of categories and complex scene compositions make it particularly suitable for evaluating model semantic scalability. To systematically investigate the relationship between interpretability and scalability in cross-attention mechanisms, we select four representative datasets with progressively increasing semantic complexity (ranging from 19 to 150 classes), simulating a continuum of segmentation difficulty.

Metrics: We use two standard metrics to evaluate semantic segmentation performance. The Mean Intersection over Union (mIoU) is the most widely adopted metric for semantic segmentation, providing a balanced measure of region similarity. It emphasizes overall region accuracy but is particularly sensitive to large objects. The Dice coefficient measures the overlap between the predicted and ground-truth regions. It is more sensitive to boundary accuracy and performs better at evaluating the segmentation of small objects than mIoU. To assess the efficiency of the parameters and computations, we define a new metric, RCPR, to compare the relative cost-performance ratio between our model and other baselines. RCPR consists of three items: Performance Retention (PR), Parameter Efficiency (PE), and Computational Efficiency (CE).

$$\mathrm{RCPR} = \mathrm{PR}^{\alpha} \times \mathrm{PE}^{\beta} \times \mathrm{CE}^{\gamma} \tag{3.28}$$

where α, β, and γ represents the importance weights. Moreover, each component is defined as:

$$\mathrm{PR} = \frac{\mathrm{mIoU}_{\mathrm{VGMEA}}}{\mathrm{mIoU}_{\mathrm{baseline}}}$$

$$\mathrm{PE} = \frac{\#\mathrm{Params}_{\mathrm{baseline}}}{\#\mathrm{Params}_{\mathrm{VGMEA}}}$$

$$\mathrm{CE} = \frac{\mathrm{GFLOP}_{\mathrm{baseline}}}{\mathrm{GFLOP}_{\mathrm{VGMEA}}}$$

Therefore, RCPR favors light-weight models with high performance and low computational burden, and RCPR > 1 defines a relatively better model than the baseline. In our experiment, we set $\alpha = 2$, $\beta = 1$, and $\gamma = 0.5$ to focus on performance and parameter efficiency rather than computational efficiency.

Training protocol: All networks are implemented with PyTorch. Specifically, CNN-based methods are reproduced with MMSegmentation [121], while ViT-based networks are reproduced with Detectron2 [702]. We employ standard data augmentation techniques, including random scaling in $[0.5, 2.0]$, random horizontal flipping, and random cropping. Training follows the standard protocols for each dataset with a batch size of 4, using the Adam-W optimizer with an initial learning rate of 0.001 and polynomial learning rate decay. For fair comparison, all models are trained under identical training protocols until convergence on 2 NVIDIA TITAN RTX GPUs.

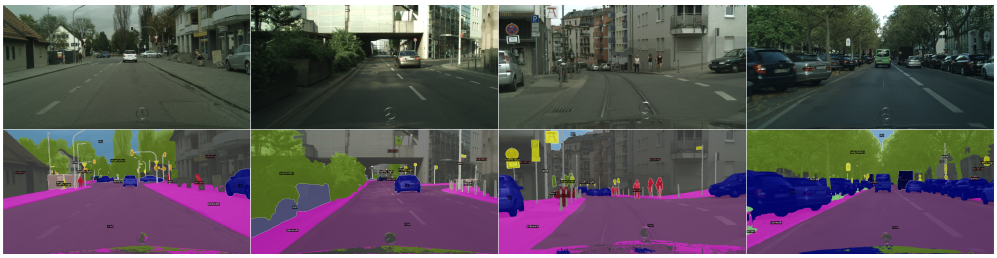
3.5.2 Performance Analysis

We evaluate our VGMEA framework against established attention mechanisms and typical semantic segmentation networks across four semantic segmentation benchmarks with increasing numbers of semantic classes. Our model is tested in three configurations: (1) VGMEA: without geometric constraints; (2) VGMEA†: with amortized variational inference for geometric constraints; and (3) VGMEA‡: with full iterative geometric constraints. We compare with two categories frameworks, CNN-based methods: Non-local[678], EncNet[806], EMANet[379], OCRNet[784], PSPNet [837], PSANet [838] and kMaxDeeplab [773]; and ViT-based approaches: MaskFormer[106], Mask2Former[105], SETR[847], HLG[813], Segmenter[595], SegFormer[712] and OneFormer[293].

Quantitative Results

Table 3.2 reports the mIoU and Dice scores on all datasets, along with #Params (parameter counts) and GFLOPS (Giga Floating-Point Operations Per Second). Our VGMEA models achieve competitive performance with significantly fewer parameters than other methods, yet consume more GLOPs.

Cityscapes: The emphasis on precise boundaries highlights the value of our TV regularization. CNN-based VGMEA‡ achieves 81.8% mIoU, outperforming all other CNN models using the fewest parameters. Moreover, ViT-based VGMEA‡ gains 82.7% mIoU, approaching to Mask2Former (84.1%) and OneFormer (84.6%) with



(a) Cityscapes successful cases

(b) Cityscapes failure cases

Figure 3.4: Visualization of semantic segmentation for the Cityscapes dataset. (a) Our model can classify pixels into the right semantic classes. (b) There are correlated semantic classes defined in the Cityscapes, resulting in the false responses between ambiguous classes, such as sidewalk vs. road vs. terrain.

significant parameter efficiency. Specifically, we exchange a 1.4% performance drop for a 27% parameter save with respect to Mask2Former, while a 1.9% drop yields a 48% save with respect to OneFormer. In addition, the 1.6% DICE improvement from w/o TV to Full TV demonstrates the critical importance of geometric constraints for improving boundary accuracy in urban scene understanding. In Fig.3.4, there are semantic segmentation results of VGMEA‡ on the Cityscapes. From the failure cases, we observe that some ambiguously defined classes in Cityscapes are the main reason for the false classification since they violate our assumption on class independence, such as bicycle/motorcycle vs. person vs. rider and sidewalk vs. road.

PASCAL Context: Our CNN-based full TV model achieves 58.4% mIoU and 59.1% Dice, which achieves the best performance over the other CNN networks. Besides, our full TV model has demonstrated competitive results in a ViT-based setting. Specifically, it achieves more than 96% of Mask2Former and OneFormer's performance with around 70% of the parameters. The amortized TV variant provides an acceptable trade-off between accuracy and computational efficiency, at the cost of more parameters in the mini-network for predicting the initial Φ surface. In Figure. 3.5, we expect that more

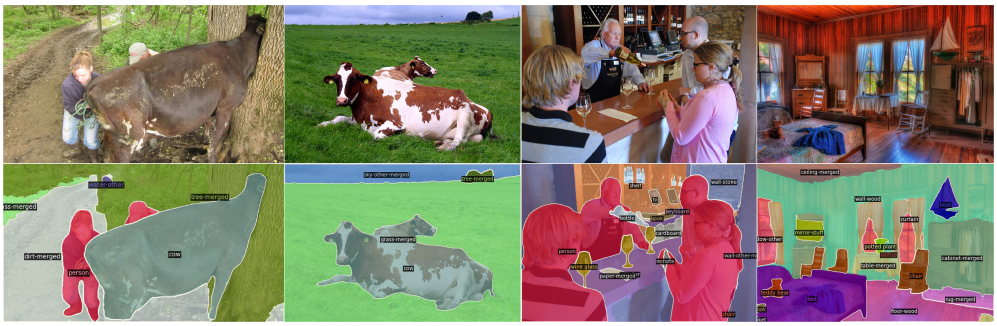
(a) PASCAL Context successful cases

(b) PASCAL Context failure cases

Figure 3.5: Visualization of semantic segmentation from the PASCAL Context dataset. (a) Our model can classify pixels into the right semantic classes. (b) There are more ambiguous semantic classes defined in the PASCAL Context, such as building vs. wall, ground vs. floor.

correlated classes have competed with each other in our feature representation space and caused more perplexed label assignment.

MSCOCO: With 81 categories, our method achieves the top performance 62.4% mIoU among CNN-based models and maintains competitive performance against the ViT-based networks. Despite the efficiency, VGMEA with Full TV achieves 96% mIoU of Mask2Former and OneFormer, and outperforms the others with fewer parameters, demonstrating remarkable scalability to complex, multi-category scenes. In addition, some classes are difficult to distinguish within limited context information, such as a backpack, suitcase, or handbag in outdoor scenarios.



(a) MSCOCO successful cases

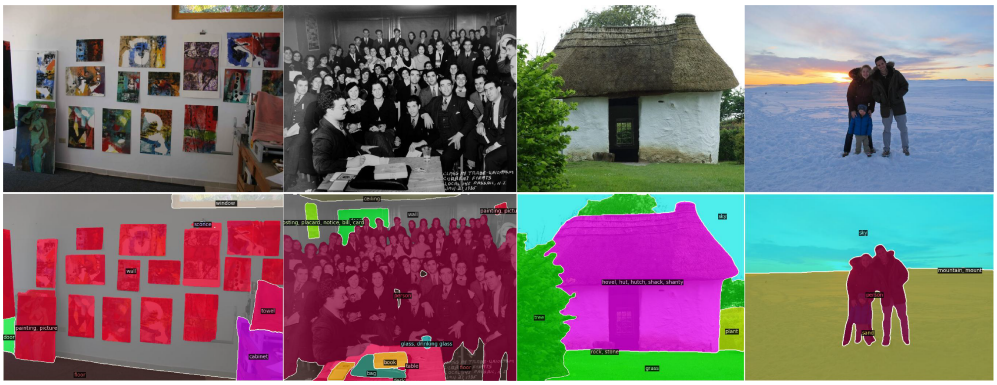
(b) MSCOCO failure cases

Figure 3.6: Visualization of semantic segmentation for the MSCOCO dataset. (a) Our model can classify pixels into the right semantic classes. (b) Some classes are difficult to distinguish from limited context information, such as backpack vs. suitcase vs. handbag.

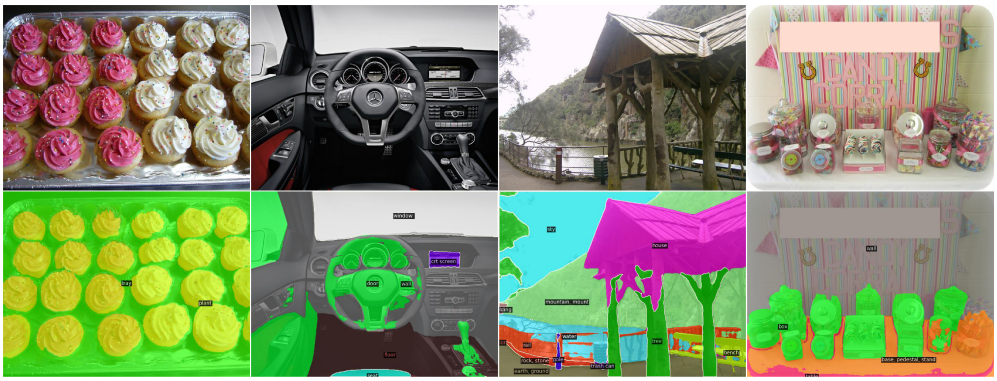
ADE20K: This dataset has diverse scenes with 150 categories, representing the most challenging scenario for model capacity. VGMEA with Full TV attains 46.5% mIoU in CNN-based setting and outperforms the other CNN networks. Moreover, the ViT-based VGMEA with Full TV (52.1%) is slightly lower than Mask2Former (53.9%) and OneFormer (54.7%), yet with around 30% fewer parameters, confirming its effectiveness across varying levels of complexity. In Figure. 3.7, our model learns the degraded feature representation space due to complicated correlation among various ambiguous classes.

Compatibility Analysis

Our VGMEA module demonstrates remarkable compatibility across diverse network architectures, serving as a lightweight drop-in replacement for conventional cross-attention



(a) ADE20K successful cases



(b) ADE20K failure cases

Figure 3.7: Visualization of semantic segmentation for ADE20K dataset. (a) Our model can classify pixels into the right semantic classes. (b) Some classes are confused by limited context information, such as house vs. tree, table vs. stand.

mechanisms without requiring architectural modifications. In CNN-based baselines, VGMEA seamlessly replaces their original attention modules with our probabilistic attention, reducing parameter count and improving performance. Meanwhile, VGMEA replaces the standard cross-attention layers in the transformer decoder for ViT-based models. As shown on the right side of Figure 3.8, VGMEA has achieved relatively better comprehensive performance than all the other benchmark models, since RCPRs are all above 1. Specifically, on the left side of the Figure 3.8, all VGMEA variants have the superiority of parameter efficiency over all the models (all PE metric values are above 1) while requiring more computational loads on average due to the bottleneck of geometrical regularization terms (CE < 1 in most cases). In addition, all VGMEA variants can achieve competitive performance compared to other benchmarks.

CNN-based	Backbone	#Params	Cityscapes 768 x 1024 (19 classes)			PASCAL Context 512 x 512 (59 classes)			MSCOCO 1024 x 1024 (80 classes)			ADE20K 512 x 512 (150 classes)		
			mIoU	Dice	GFLOPs	mIoU	Dice	GFLOPs	mIoU	Dice	GFLOPs	mIoU	Dice	GFLOPs
NonLocal[678]	ResNet-101	67	78.7	79.2	911	45.4	46.2	286	52.4	53.2	1249	42.9	43.6	287
EncNet[806]		53	78.6	79.4	656	51.7	52.8	219	58.1	58.3	876	42.1	43.2	219
EMANet[379]		59	79.1	79.6	742	53.1	54.8	247	59.4	59.8	990	43.3	44.1	247
OCRNet[784]		56	80.1	80.4	693	56.2	57.3	231	60.4	61.1	926	43.9	44.5	231
kMaxDeeplab[773]		73	79.7	80.2	1004	57.8	58.4	320	60.7	61.0	1468	45.3	46.2	324
PSANet[838]		76	78.4	78.6	843	46.2	46.4	279	58.4	58.6	1107	43.8	44.3	274
PSPNet[837]		66	79.4	79.8	769	47.8	48.4	256	58.7	59.1	1021	43.6	43.9	257
VGMEA		42	80.6	80.9	648	56.4	57.1	250	60.8	61.4	982	45.8	46.9	254
VGMEA†		44	81.2	81.8	728	57.2	58.1	319	61.3	61.7	1128	46.1	47.4	324
VGMEA‡		**42**	**81.8**	**82.5**	995	**58.4**	**59.1**	358	**62.4**	**62.5**	1367	**46.5**	**48.7**	367
Transformer-based	**Backbone**	**#Params**	**Cityscapes 768 x 1024 (19 classes)**			**PASCAL Context 512 x 512 (59 classes)**			**MSCOCO 1024 x 1024 (80 classes)**			**ADE20K 512 x 512 (150 classes)**		
			mIoU	Dice	GFLOPs	mIoU	Dice	GFLOPs	mIoU	Dice	GFLOPs	mIoU	Dice	GFLOPs
MaskFormer[106]	ViT-Base	112	81.4	82.3	1027	62.3	63.1	584	64.8	65.1	1378	49.4	50.1	571
Mask2Former[105]		128	**83.5**	84.1	1385	65.7	66.4	792	**67.4**	**68.6**	1712	53.9	54.3	804
SETR[847]		160	79.2	79.8	1215	56.4	57.2	687	64.3	64.9	1559	47.7	48.1	688
HLG[813]		100	79.5	80.6	1183	57.5	58.1	618	64.1	64.3	1498	46.2	46.9	621
OneFormer[293]		180	83.1	**84.6**	1370	**66.8**	**67.1**	684	**67.4**	68.4	1690	**54.7**	**55.2**	678
Segmenter[595]		102	80.6	81.4	1205	59.7	60.4	563	62.3	63.1	1625	48.1	48.6	548
SegFormer[712]		129	82.3	83.6	1148	62.1	63.3	572	62.7	63.4	1632	49.6	50.4	568
VGMEA		**91**	81.3	81.6	1150	61.2	62.7	678	63.1	63.3	1477	50.8	51.2	690
VGMEA†		100	82.2	82.4	1320	62.8	63.2	775	63.8	64.1	1674	51.7	51.9	782
VGMEA‡		93	82.7	82.9	1520	63.9	64.5	850	64.7	64.9	1845	52.1	52.3	832

Table 3.2: Performance comparison on four datasets. Parameter counts are in millions (M), and GFLOPs denotes gigaflops. The best results are in bold, and the second-best results are underlined. † means using amortized TV (with 3 iterations), and ‡ represents the full TV (with 15 iterations).

In detail, we substitute the encoding layer in EncNet with our light-weighted module, reducing the parameter count from 53M to 42M while maintaining consistent gains in mIoU. Similarly, in EMANet, VGMEA replaces the expectation-maximization attention, achieving comparable performance (105% PR) with higher parameter efficiency (140% PE). For ViT-based models like Mask2Former and OneFormer, VGMEA replaces the standard cross-attention layers in the transformer decoder. This replacement is particularly effective because our method is inherently compatible with query-based mechanisms. VGMEA maintains 98% PR of the original mIoU while increasing PE to 128% and 180%, respectively. In conclusion, the universal compatibility stems from VGMEA's adherence to standard cross-attention interfaces (query-key-value inputs and residual outputs), making it suitable for integration into any modern segmentation pipeline. Besides, our geometrical regularization can generally increase the performance, but at the cost of a high computational burden.

EM iters	mIoU (%)	Time (ms)
1	51.5	0.8
2	51.6	1.1
3	51.6	1.4
4	51.8	1.6
5	52.1	1.8

Table 3.3: Ablation study on the effects of EM iterations against the inference performance and speed. Experiments are conducted on ViT-Base with the ADE20K VOC dataset.

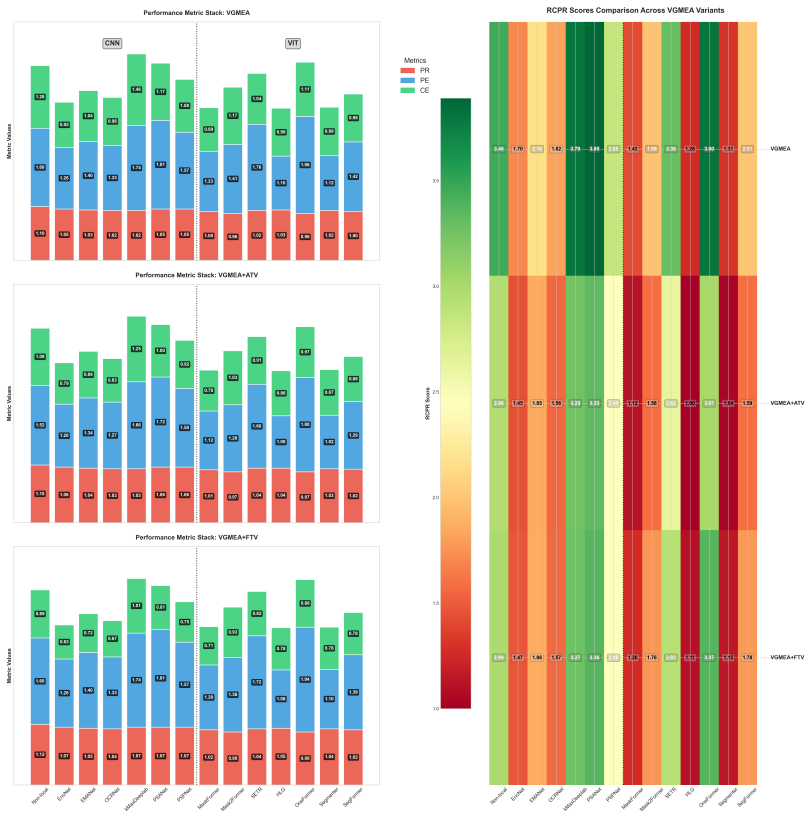


Figure 3.8: Cross-architecture compatibility analysis for three VGMEA variants: VGMEA uses no TV, VGMEA+ATV uses amortized TV, and VGMEA+FTV uses the full TV. PE, PR, CE and RCPR represent **P**arameter **E**fficiency (PE), **P**erformance **R**etention (PR), **C**omputational **E**fficiency (CE) and **R**elative **C**ost-**P**erformance **R**atio (CPR), respectively. For all metrics, values above 1 are better.

Convergence Analysis

On the one hand, we empirically analyze EM convergence in terms of training stability, given the theoretical guarantees of the EM algorithm. On the other hand, we estimate the inference efficiency in terms of EM iterations.

Training Stability: Due to the theoretical guarantees of the EM algorithm, the

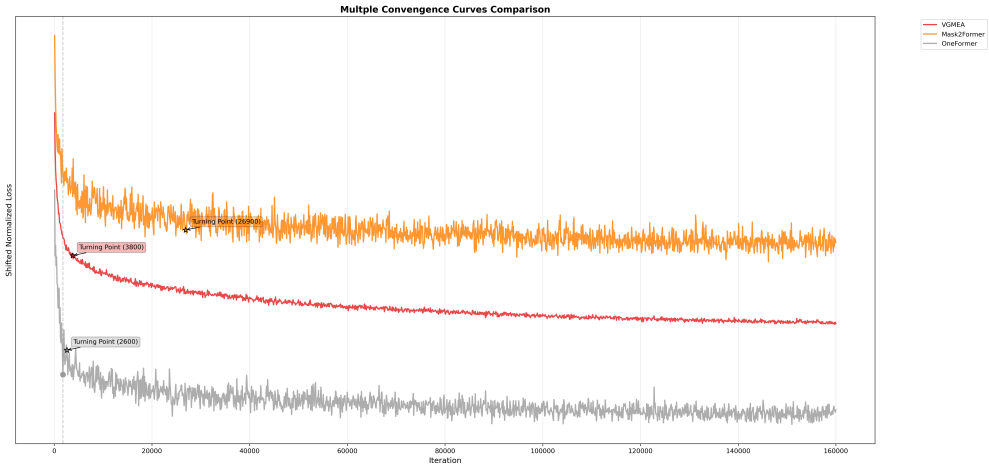


Figure 3.9: Ablation study on the convergence analysis of VGMEA against stacked cross-attention in Mask2Former and OneFormer on the ADE20K dataset.

VGMEA typically converges within 3-5 iterations via the evidence lower bound (ELBO) in a mini-batch. Compared to a deep stack of traditional attention blocks (i.e., 6-12 layers in OneFormer), VGMEA demonstrates smooth convergence curves compared to the oscillatory behavior sometimes observed in deep stacks (Figure 3.9).

Inference Efficiency: Due to the property of the moment-based algorithm, VGMEA can vary the EM iterations in the inference phase without changing the computational graph. That means fewer EM iterations are required for inference without significantly reducing performance, unlike fixed deep stack architectures, where inference typically passes through as many layers as in the training stage. This flexibility is particularly valuable for real-time applications. As shown in the Table. 3.3, we can run the EM algorithm once to achieve fast inference speed while maintaining good performance.

Interpretability Analysis

From a Gaussian mixture model perspective, VGMEA offers straightforward interpretability in the feature space during both the learning and inference stages.

During training, VGMEA learns to form the feature space, where each Gaussian component $\mathcal{N}(\boldsymbol{\mu}_k, \mathbf{C}_k)$ evolves to represent a semantic concept. The centroids $\boldsymbol{\mu}_k$ converge to prototypical feature vectors for each category, while the precision matrices $\mathbf{C}_k^{-1}$ capture intra-class variability. The orthogonal constraint ensures that different

concepts occupy nearly independent subspaces, facilitating clear semantic separation. Meanwhile, during inference, the responses of the feature space $r_{kn} = q(z_n = k)$ provide probabilistic assignments of each pixel to semantic concepts. The fact offers several interpretability advantages: (1) uncertainty quantification measured by classification confidence given by the posterior distribution; (2) concept attention visualized by r_{kn} activated in different regions; (3) representation disentanglement that analyzes the personalized input-dependent details separated from the common class-wise features; (4) ontology analysis that can detect the ambiguous semantic conception design in datasets.

3.5.3 Ablation Study

To thoroughly understand the contributions of each component in our VGMEA framework, we conduct comprehensive ablation studies analyzing the individual and combined effects of our key innovations.

Amortized Geometric Constraints

We first investigate the impact of our amortized total variation (TV) regularization scheme, a crucial innovation that balances geometric consistency with computational efficiency.

Experimental Setup: We compare five configurations: (a) No geometric constraints, (b) Full iterative TV regularization (15 iterations), and (c)-(e) Our amortized TV approach ($t \leq 5$ iterations + prediction network). Evaluations are conducted on all four datasets using the DICE metric to highlight the precise delineation of boundaries. From Table 3.4, we observe that the general capability of preserving effectiveness. On average, the amortized TV variants retain more than 90% of the full TV's boundary delineation while reducing the inference time. In detail, the more iterations VGEMA performs, the higher its DICE score, but at the cost of more inference time. The benefits are consistent across datasets, with similar relative improvements observed across all four. Compared to the VGMEA with no TV terms (decrease $\geq -4.2\%$), the amortized TV with 5 iterations has retained the highest DICE score ($\geq 98.2\%$) but has spent more computation time on level-set evolution ($\geq +1.1$ ms per image). Although they do not completely avoid the TV bottleneck, amortized variants can be a good compromise between performance and inference speed.

Configuration	Cityscapes val set		
	DICE (%)	Time/Image (ms)	Performance Gap
Full TV (15 iter)	82.9	2.5	0 (baseline)
No Constraints	78.3	0.5	-4.6% / -2.0ms
Amortized TV (1 iter)	79.1	0.9	-3.8% / -1.6ms
Amortized TV (3 iter)	80.2	1.5	-2.7% / -1.0ms
Amortized TV (5 iter)	81.1	1.8	-1.8% / -0.7ms

Configuration	PASCAL Context val set		
	DICE (%)	Inference Time (ms)	Performance Gap
Full TV (15 iter)	63.6	2.3	0 (baseline)
No Constraints	59.4	0.6	-4.2% / -1.7ms
Amortized TV (1 iter)	61.2	0.8	-2.4% / -1.5ms
Amortized TV (3 iter)	62.8	1.4	-0.8% / -0.9ms
Amortized TV (5 iter)	62.8	1.7	-0.8% / -0.6ms

Configuration	MSCOCO val set		
	DICE (%)	Time/Image (ms)	Performance Gap
Full TV (15 iter)	64.9	3.8	0 (baseline)
No Constraints	60.7	0.9	-4.2% / -2.9ms
Amortized TV (1 iter)	60.9	1.6	-4.0% / -2.2ms
Amortized TV (3 iter)	62.7	2.3	-2.2% / -1.5ms
Amortized TV (5 iter)	63.3	2.9	-1.6% / -0.9ms

Configuration	ADE20K val set		
	DICE (%)	Time/Image (ms)	Performance Gap
Full TV (15 iter)	52.3	5.4	0 (baseline)
No Constraints	44.9	1.2	-7.4% / -4.2ms
Amortized TV (1 iter)	48.2	3.1	-4.1% / -2.3ms
Amortized TV (3 iter)	49.5	3.9	-2.8% / -1.5ms
Amortized TV (5 iter)	50.8	4.8	-1.5% / -0.6ms

Table 3.4: Ablation study on geometric constraints on all datasets.

Orthogonal Constraint

The orthogonal constraint on concept centroids serves multiple purposes: improving interpretability, enabling computational simplifications, and enhancing semantic separation.

Experimental Setup: We evaluate four conditions: (a) No constraints, (b) Orthogonality only, (c) Diagonal covariance only, and (d) Full orthogonal + diagonal constraints. Measurements include parameter efficiency, semantic separation quality, and computational benefits.

Inter-class Separation quantifies the average dissimilarity between different semantic

concept centroids in the learned feature space. It measures how well the model distinguishes between different categories. Let $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ represent the centroid vectors of concepts i and j respectively, where $i, j \in 1, 2, ..., K$ and $i \neq j$. The Inter-class Separation $\mathcal{S}_{\text{inter}}$ is defined as:

$$S_{\text{inter}} = \frac{2}{K(K-1)} \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \Big(1 - \text{cosine}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j)\Big) \tag{3.29}$$

Intra-class Coherence measures the average compactness of features within each semantic concept. It assesses how tightly clustered the features are around their respective concept centroids. For each concept k, let $\mathcal{X}_k = \mathbf{x}_n | z_n = k$ be the set of feature vectors assigned to concept k. The Intra-class Coherence $\mathbf{C}_{\text{intra}}$ is defined as:

$$\mathbf{C}_{\text{intra}} = \frac{1}{K} \sum_{k=1}^{K} \left(\frac{1}{|\mathcal{X}_k|} \sum_{\mathbf{x}_n \in \mathcal{X}_k} \text{cosine}(\mathbf{x}_\text{n}, \boldsymbol{\mu}_\text{k}) \right) \tag{3.30}$$

Where the cosine similarity is calculated as:

$$\text{cosine}(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j) = \frac{\boldsymbol{\mu}_i \cdot \boldsymbol{\mu}_j}{\|\boldsymbol{\mu}_i\| \cdot \|\boldsymbol{\mu}_j\|} \tag{3.31}$$

Therefore, two metrics provide quantitative evidence for the effectiveness of our orthogonal constraints in creating a well-structured feature space that supports robust semantic segmentation. Specifically, high $\mathcal{S}_{\text{inter}}$ and $\mathbf{C}_{\text{intra}}$ values correlate with better decision boundaries, reduced classification ambiguity, improved generalization performance, and enhanced model interpretability.

We conclude from Table 3.5 that on average orthogonality + diagonal correlation enhances the interpretability (improves inter-class separation $0.39 \rightarrow 0.85$), reduces the parameter count (18.1% by eliminating off-diagonal covariance elements), and accelerates the inference time (by 46.1% due to simplified matrix operations), but may degrade the semantic segmentation precision. That reveals a dilemma: explainable modules are likely to sacrifice performance, especially when the concept designs are ambiguous or complex in the dataset (i.e., terrain vs. sidewalk). Therefore, more prior information needs to be integrated into the design of explainable modules, for instance, a tree-structured conception in which the parent nodes represent the super-categories.

Configuration	Cityscapes val set				
	mIoU (%)	Params (M)	Inter-class Separation ↑	Intra-class Coherence ↑	Inference Time (ms)
None	83.2	48.4	0.48	0.44	4.2
Orthogonality	82.8	48.4	0.89	0.85	4.2
Diagonal Correlation	82.9	42.1	0.54	0.45	2.3
Full Constraints	83.1	42.2	0.88	0.87	2.5
Configuration	PASCAL Context val set				
	mIoU (%)	Params (M)	Inter-class Separation ↑	Intra-class Coherence ↑	Inference Time (ms)
None	64.1	48.7	0.45	0.46	4.5
Orthogonality Only	63.1	48.7	0.88	0.87	4.5
Diagonal Correlation	62.7	42.2	0.52	0.51	2.4
Full Constraints	63.9	42.2	0.87	0.88	2.2
Configuration	MSCOCO val set				
	mIoU (%)	Params (M)	Inter-class Separation ↑	Intra-class Coherence ↑	Inference Time (ms)
None	65.3	52.7	0.36	0.38	5.1
Orthogonality Only	62.3	52.6	0.84	0.82	5.0
Diagonal Correlation	62.8	42.4	0.48	0.44	2.9
Full Constraints	64.7	42.5	0.83	0.82	2.8
Configuration	ADE20K val set				
	mIoU (%)	Params (M)	Inter-class Separation ↑	Intra-class Coherence ↑	Inference Time (ms)
None	53.1	57.3	0.28	0.31	5.9
Orthogonality Only	51.8	57.4	0.82	0.83	5.8
Diagonal Correlation	51.7	42.6	0.31	0.32	3.2
Full Constraints	52.6	42.6	0.83	0.84	3.1

Table 3.5: Ablation study on orthogonal constraints on all datasets

Trade-off between efficiency and accuracy

We analyze the overall efficiency-accuracy characteristics of different VGMEA configurations to provide practical guidance for deployment scenarios. There are 6 configurations formed by combinations of 3 TV options (none, amortized, full) with 2 orthogonal constraint options (none, orthogonality).

From Table 3.6, we can obtain optimal configurations for different scenarios on average: (a) the best efficiency chooses No TV + orthogonality for the highest parameter efficiency (60.5% mIoU vs. 40M Param) and speed (2.6 ms per image); (b) best balance adopts amortized TV + orthogonality that offers the optimal balance, achieving 97.2% of best performance with 64.0% of inference speed; (c) best interpretability combines full TV and orthogonality that achieves the second best performance but the complete semantic concept separation; (d) best Accuracy uses full TV + no orthogonality reach peak performance for interpretability-neglect applications. In conclusion, we observe cross-dataset consistency: the configuration of amortized TV plus orthogonality consistently provides the best trade-off between efficiency and accuracy, regardless of dataset complexity or characteristics.

Summary of Ablation Findings

Configuration	Cityscapes val set				
	mIoU (%)	Params (M)	GFLOPs	Memory (MB)	Inference Speed (ms)
No TV, No Orthogonal	79.2	48	485	2843	3.4
No TV, Orthogonal	79.4	40	324	2346	2.6
Amort TV, No Orthogonal	80.6	52	824	3956	4.7
Amort TV, Orthogonal	80.4	48	648	2748	3.5
Full TV, No Orthogonal	81.6	46	1230	3078	5.6
Full TV, Orthogonal	81.3	42	995	2578	4.3
Configuration	PASCAL Context val set				
	mIoU (%)	Params (M)	GFLOPs	Memory (MB)	Inference Speed (ms)
No TV, No Orthogonal	60.1	48	264	2975	3.5
No TV, Orthogonal	59.3	40	217	2453	2.5
Amort TV, No Orthogonal	62.4	50	397	4173	4.8
Amort TV, Orthogonal	62.7	45	319	2875	3.4
Full TV, No Orthogonal	64.1	48	537	3248	5.7
Full TV, Orthogonal	63.5	42	358	2671	4.4
Configuration	MSCOCO val set				
	mIoU (%)	Params (M)	GFLOPs	Memory (MB)	Inference Speed (ms)
No TV, No Orthogonal	60.3	47	1246	3247	3.8
No TV, Orthogonal	59.4	40	1078	2978	2.9
Amort TV, No Orthogonal	61.9	52	1578	4524	5.7
Amort TV, Orthogonal	61.3	48	1429	3314	4.3
Full TV, No Orthogonal	63.8	50	1684	3745	6.8
Full TV, Orthogonal	62.4	42	1367	3126	5.5
Configuration	ADE20K val set				
	mIoU (%)	Params (M)	GFLOPs	Memory (MB)	Inference Speed (ms)
No TV, No Orthogonal	44.3	49	287	3879	4.2
No TV, Orthogonal	43.8	40	224	3598	3.3
Amort TV, No Orthogonal	45.9	54	425	4975	6.8
Amort TV, Orthogonal	45.4	50	324	4015	5.1
Full TV, No Orthogonal	47.3	52	512	4178	7.4
Full TV, Orthogonal	46.5	42	367	3785	5.8

Table 3.6: Trade-off between efficiency and accuracy analysis on all datasets.

Our ablation studies demonstrate that:

1. Amortized TV provides 97.2% of full TV's benefits with 64.0% faster inference speed.

2. Orthogonal constraints improve semantic separation by 46.0% while reducing parameters by 18.1%.

3. The optimal configuration (Amortized TV + Orthogonality) achieves the best efficiency-accuracy balance across all datasets.

4. Our approach offers a flexible configuration suitable for diverse deployment scenarios from real-time to accuracy-critical applications.

These results validate the design choices in VGMEA and provide practical guidance for researchers and practitioners seeking to leverage our method in various applications.

3.5.4 Limitations

Despite the compelling advantages of our VGMEA framework, several limitations merit careful consideration and suggest promising directions for future research.

Limited Feature Space Adaptability of EM Iteration

The Expectation-Maximization algorithm, while providing theoretical guarantees and stable convergence, operates within a fundamentally static feature space framework. Unlike deep cross-attention stacks that enable hierarchical feature transformation through multiple layers, our EM-based approach performs iterative refinement within a fixed feature representation space. This limitation manifests in two critical aspects.

First, the inability of feature space migration restricts the modules evolving from abstract semantic concepts to dataset-specific instantiations. While traditional attention mechanisms can progressively transform and specialize features through stacked layers (e.g., $\mathbf{Q}^{(l+1)} = f(\mathbf{Q}^{(l)}, \mathbf{K}, \mathbf{V})$), our EM iterations primarily optimize assignment probabilities and centroid positions within the original feature space:

$$\boldsymbol{\mu}_k^{(t+1)} = \boldsymbol{\mu}_k^{(t)} + \eta \frac{\sum_n r_{kn}^{(t)} (\mathbf{x}_n - \boldsymbol{\mu}_k^{(t)})}{\sum_n r_{kn}^{(t)}}$$

Second, this static feature space assumption may limit generalization across diverse domains. As shown in Table 3.7, while VGMEA performs excellently on structured datasets such as Cityscapes, its performance gain diminishes on more diverse datasets such as ADE20K, where adaptive feature transformation becomes crucial.

In summary, the practical implications are (a) limited hierarchical abstraction that VGMEA operates at a single level of abstraction while cross-attention stacks can progressively abstract features from local patterns to global semantics; (b) reduced adaptability that the fixed feature space may struggle with domain shifts; (c) architectural rigidity that inability to learn the complex feature transformations hinders deep transformers in few-shot. We suggest that future work could explore hybrid approaches where EM iterations are interleaved with feature transformation layers, creating a dynamic feature space that evolves during optimization.

Semantic Over-Constraint through Orthogonality

Dataset	Semantic Classes	VGMEA	Stacked Cross-Attention	Performance Gap
Cityscapes	19	82.7	83.1	-0.4%
PASCAL	59	63.9	66.8	-2.9%
MSCOCO	80	64.7	67.4	-2.7%
ADE20K	150	52.1	54.7	-2.6%

Table 3.7: Performance comparison highlighting feature adaptability limitations via mIoU(%). Our model adopts full TV + orthogonality while the stacked cross-attention modules are from OneFormer.

The orthogonal constraint $\boldsymbol{\mu}_i^\top \boldsymbol{\mu}_j \approx 0$ for $i \neq j$, while computationally beneficial, imposes an artificial independence assumption that contradicts real-world semantic relationships. This limitation is particularly evident in natural semantic hierarchies that often exhibit inherent relationships (e.g., "vehicle" $\rightarrow$ "car" $\rightarrow$ "wheel") that should be reflected in the feature space geometry. The orthogonal constraint forces these related concepts to occupy perpendicular subspaces, discarding valuable relational information. Additionally, the computational savings from diagonal covariance matrices come at the cost of ignoring informative feature correlations that could enhance segmentation accuracy. In addition, the constraint leads to inefficient utilization of the feature space. To maintain orthogonality among K concepts in D-dimensional space, we require $D \geq K$, potentially forcing expansion into higher dimensions than necessary if some natural dependence were permitted.

Our experiments show that while orthogonality improves separation metrics, the performance gains diminish as the number of concepts increases, especially when concepts are complex or ambiguous. That suggests increasing "semantic tension" as category relationships become more complex. We advise that future research can focus on hierarchical semantic structures to accommodate layer-wise orthogonal constraints. The architectures might preserve computational benefits while accommodating complex semantic relationships. Specifically, a bottom-top structure consists of base-level orthogonality that enforces orthogonality only between semantically distant categories, hierarchical dependence that allows dependent relationships along category hierarchies, and conditional independence that maintains discriminability among categories on different branches.

Computational Bottleneck of TV Regularization

The Total Variation regularization, while effective for boundary preservation, suffers from significant computational challenges that limit practical deployment. The absence

of a closed-form solution necessitates iterative optimization through level-set methods. Each iteration requires computation of the curvature:

$$\kappa = \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right)$$

This process typically requires more than 15 iterations for convergence, accounting for around 50% of total computation time. Furthermore, the level-set evolution approach used for TV minimization is inherently sequential and difficult to parallelize efficiently on modern hardware. We suggest addressing this bottleneck with several promising approaches: (a) train lightweight networks as a quick TV solver to predict optimal TV regularization directly from features; (b) apply multi-grid Methods to accelerate convergence; (c) use learned priors to provide better predictive initialization, reducing convergence iterations.

3.6 Conclusions

In this chapter, we have presented VGMEA, a novel variational Gaussian mixture model embedded attention framework that fundamentally reinterprets cross-attention mechanisms through a probabilistic lens. Our approach establishes a mathematically grounded foundation for attention operations by formulating them as iterative concept distillation processes within a mean-shift framework. The theoretical innovation can unify diverse attention variants under a single probabilistic framework. Through extensive experiments and ablation analysis, we conclude that our VGMEA not only enhances theoretical understanding but also provides practical adaptation to deep learning approaches. It pioneers a viable pathway for integrating sophisticated probabilistic models into deep learning architectures, bridging the long-standing gap between Bayesian methods and modern neural networks through innovations in online Bayesian updates and amortized variational inference. Despite these advancements, we acknowledge several limitations of our current approach on feature space transformation, semantic over-constraint on orthogonality, and the bottleneck of the Total Variation solver. Therefore, we suggest future work to integrate hierarchical orthogonal constraints and develop more efficient solvers for geometrical regularization. In conclusion, we believe that VGMEA represents a significant step toward more interpretable, efficient, and theoretically sound attention mechanisms, and we hope it will inspire further research at the intersection of probabilistic modeling and deep learning for computer vision and

beyond.