Improving CNN Generalization with PDE Preprocessing and the Variational Information Bottleneck

Gor Gharagyozyan
Institute for Informatics and Automation
Problems of NAS RA
Yerevan, Armenia
e-mail: gor.gharagyozyan@edu.isec.am

***Abstract*—This paper presents a hybrid approach that integrates pre-defined convolutional layers based on Partial Differential Equations (PDEs) with the Variational Information Bottleneck (VIB) framework to improve image classification performance. While PDE-based layers enhance low-level structural features in the input, they may also introduce redundant information. To address this, we introduce a VIB module after the PDE layers, which learns compressed latent representations that retain only task-relevant information. The proposed architecture is evaluated on the CIFAR-10 dataset using various CNN backbones, including ResNet and VGG. Experimental results show that our method improves classification accuracy while reducing representational complexity. This demonstrates that combining physics-inspired priors with information-theoretic compression offers an effective strategy for enhancing deep neural networks.**

***Keywords*—Information bottleneck, partial differential equations, deep learning, image classification, representation compression.**

I. Introduction

Convolutional Neural Networks (CNNs) have achieved remarkable success in image classification by learning hierarchical representations from raw data. While deep architectures such as ResNet [1] and VGG [2] effectively capture complex patterns, their early layers often rely on data-driven training to extract low-level visual features such as edges and textures. Recent studies [3] have shown that incorporating domain knowledge, particularly in the form of Partial Differential Equation (PDE)-based filters, can enhance early representations by embedding structural priors into the network. PDE-based convolutional layers offer a physics-inspired alternative to learned filters, improving generalization without increasing model complexity. However, these pre-defined transformations may also preserve redundant or task-irrelevant information, potentially propagating noise deeper into the network. This motivates the need for a principled mechanism to control the flow of information extracted by such filters. To address this, we propose a hybrid architecture that combines PDE-based convolutional preprocessing with the Variational Information Bottleneck (VIB) framework [4]. The VIB module, positioned after the PDE layers, learns a stochastic latent representation that compresses irrelevant details while preserving task-relevant information. By integrating physics-based priors with information-theoretic compression, the proposed method aims to improve classification performance and representation efficiency.

II. Theoretical Foundations

2.1 PDE-Based Convolutional Layers

The early-stage representation is computed using fixed filters derived from discretized Partial Differential Equations (PDEs). For example, the 2D heat equation (parabolic PDE) is expressed as:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial t^2} + \frac{\partial^2 u}{\partial y^2}$$

This is discretized via finite differences [5] into:

$$u_{i,j}^{t+1} = u_{i,j}^{t} + \varphi \times P(u^t)$$

where P is a convolution operator resembling the Laplacian kernel and φ is a fixed scaling parameter.

Similarly, the hyperbolic variant is derived from the 2D wave equation and includes second-order temporal differences.

2.2 VIB

The core idea of VIB is to learn a representation T of the input X that keeps only the information needed to predict the output Y and forgets everything else.

In its classical formulation, the Information Bottleneck [6] principle seeks to optimize the trade-off between compression and predictive relevance by minimizing the following functional

$$I(X;T) - \beta I(T;Y)$$

where X is the input variable that we want to compress (e.g., image), T is the compressed representation of X, Y is the label (e.g., class), I is the mutual information between two variables [7]

$$I(X;T) = H(T) - H(T/X)$$

and β is a trade-off parameter.

We want T to be:

- **Compact** (low $I(X;T)$): throw away noise and irrelevant information.

https://doi.org/10.51408/csit2025_35

- **Useful** (high $I(T;Y)$): retain what helps predict Y.

The (VIB) reformulates the above with a neural encoder and decoder, inspired by variational autoencoders [8].

Encoder converts the given representation into a distribution $q(t\ /\ x)$ (typically modeled as a Gaussian) [4]:

$$q(t\ /\ x)\ =\ N(t\ /\ \mu(x),\ \sigma^2(x))$$

This allows T to be stochastic, which introduces noise and forces compression. The idea is to discourage the model from memorizing everything and instead train it to extract a probabilistic summary of the input that preserves only task-relevant information.

Decoder predicts label y from sampled latent t, and typically a small neural classifier.

The training objective of our model, as described in [4], will be:

$$\mathcal{L}_{VIB} = \mathrm{E}_{p(x,y)}\Big[\mathrm{E}_{q(t\ /\ x)}[-\log p\ (y\ /\ t)]\Big] +$$

$$\beta \cdot D_{KL}\big(q(t\ /\ x)\ \|\ p(t)\big)$$

where the first term is the classification loss (cross-entropy under stochastic encoding), the second term is the compression loss: penalizes how much $q(t\ /\ x)$ diverges from a simple prior $p(t)$, and β is the coefficient showing how much you value compression. The KL divergence

$$D_{KL}(P\ \|\ Q)\ =\ \sum_{x\in X} P(x)\log\ \ \frac{P(x)}{Q(x)}$$

term [7] encourages the encoder $q(t\ /\ x)$ to be as uncertain as possible about irrelevant info, but precise about what helps predict Y.

General comparison between VIB and Classical CNN is illustrated in Table 1.

Aspect	Classical CNN	VIB
Representation	Deterministic	Stochastic
Regularization	Dropout / BatchNorm	Information-theoretic (KL)
Goal	Minimize CE loss	Balance CE + compression
Interpretability	Low	High (info plane analysis)

Table 1. VIB vs Classical CNN

The summary of VIB intuition is to learn a compressed, uncertain, task-relevant representation of your input by letting the model choose what to keep and what to forget.

III. Experimental Setup and Observations

To evaluate the proposed approach, we conducted experiments on the CIFAR-10 dataset, which consists of 60,000 color images across 10 classes. The focus of the experiments was to assess how integrating a Variational Information Bottleneck (VIB) module after fixed PDE-based convolutional layers affects the performance of standard CNN architectures.

The experiments also explored how different values of the VIB regularization parameter β β affect the trade-off between compression and accuracy. In all cases, models were trained from scratch using identical optimization settings to ensure a fair comparison.

3.1 Architecture Design

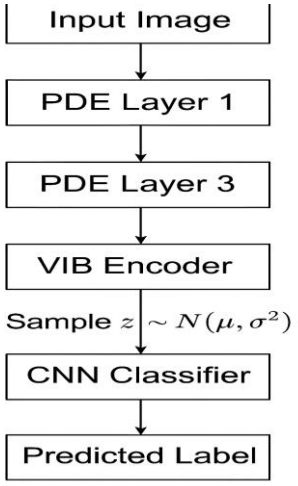

Fig. 1. Overall architecture of the proposed PDE-VIB-CNN model

The network is composed of three main stages, as illustrated in Figure 1:

1. *PDE-based Convolutional Layer Stack*: The first part of the network consists of three learnable convolutional layers derived from the discretization of parabolic partial differential equations. These PDE-inspired layers encode structural priors such as local smoothness and edge continuity, while retaining the flexibility of trainable weights. Each of them is followed by a ReLU activation and batch normalization.
2. *VIB Module*: The output of the PDE layers is passed to a fully connected encoder network that produces a mean vector $\mu(x) \in \mathbb{R}^d$ and a log-variance vector $log\sigma^2(x) \in \mathbb{R}^d$. These define a conditional Gaussian distribution over a latent variable z, from which samples are drawn using the reparameterization trick. This stochastic bottleneck enforces an information constraint, allowing the model to retain only the most task-relevant features from the PDE-enhanced representation.
3. The sampled latent variable t is then passed to a standard CNN backbone such as ResNet18 or VGG11, which performs the final classification. The entire network is trained end-to-end using a composite loss function that includes both cross-entropy for label prediction and a KL divergence term that penalizes deviations from a unit Gaussian prior on t ..

This modular structure: PDE layers for low-level priors, VIB for information compression, and CNN for classification, offers a balanced trade-off between structured feature engineering and learned abstraction.

3.2 Training Strategy and Setup

All models were implemented in *PyTorch (*an open-source deep learning framework) [9] and trained using stochastic gradient descent with momentum.

Data augmentation techniques such as random horizontal flipping and cropping were used to improve generalization.

The VIB regularization coefficient β was tuned empirically to balance compression and classification accuracy. Batch normalization and dropout were employed to stabilize training and reduce overfitting, especially in the deeper layers of the CNN.

3.3 Information Flow and Optimization

Once the input is encoded by the PDE-based layers and passed through normalization and activation functions, the VIB module receives this enriched representation and performs probabilistic compression. Rather than repeating the deterministic flow seen in typical CNNs, here the encoder outputs two functions of the input — a mean vector $\mu(x)$ and a log-variance vector $\log \sigma^2(x)$, which define a conditional Gaussian distribution over the latent variable t. Sampling from this distribution is done via the reparameterization trick:

$$t = \mu(x) + \sigma(x) \odot \varepsilon$$

where $\odot$ means element-wise multiplication and ε is a random noise vector sampled from the standard multivariate normal distribution $N(0, I)$ with zero mean and identity covariance.

During backpropagation, the KL divergence term in the loss $\mathcal{L}_{VIB}$ penalizes deviation from a simple prior. This not only encourages compression but also stabilizes training by preventing overfitting. As a result, the model learns to emphasize task-relevant patterns while suppressing noisy or redundant information that might be passed from the PDE layers.

3.4 Qualitative Observations

In our experiments, the inclusion of the VIB module led to consistently more compact latent representations, as measured by lower KL divergence values and improved calibration of class probabilities. Models with PDE preprocessing alone showed moderate gains over standard CNNs, while the addition of VIB further enhanced the network's ability to focus on discriminative information. Training was also observed to be more stable in the presence of the VIB layer, likely due to its regularizing effect on feature flow.

Overall, the combination of physically motivated preprocessing and information-theoretic compression yielded improvements in representation quality, robustness to noise, and generalization behavior. These benefits were observed consistently across both ResNet and VGG backbones.

IV. Conclusion

In this work, we proposed a hybrid neural architecture that integrates trainable PDE-based convolutional layers with the VIB framework. By combining the inductive bias of physics-inspired filtering with information-theoretic regularization, our approach encourages the extraction of compact and task-relevant representations. The VIB module acts as a stochastic bottleneck, effectively limiting information flow to the classifier and improving robustness to irrelevant features.

Preliminary results demonstrate that the proposed method is feasible and structurally compatible with existing CNN backbones. Moreover, its components remain fully differentiable and end-to-end trainable, making the architecture suitable for deployment in low-data or noisy environments. Future work will involve comprehensive experimentation, ablation studies, and an exploration of alternative PDE formulations.

Acknowledgment

The author would like to express his sincere gratitude to Prof. Mariam Haroutunian for her valuable guidance, insightful feedback, and continuous support throughout the development of this research.

References

[1] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 770–778, 2016.

[2] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition", *arXiv:1409.1556, 2014.*

[3] V. R. Sahakyan, V. G. Melkonyan, G. A. Gharagyozyan and A. S. Avetisyan, "Enhancing Image Recognition with Pre-Defined Convolutional Layers Based on PDEs", *Programming and Computer Software*, vol. 49, no. 3, pp. 192-197, 2023.

[4] A. A. Alemi, I. Fischer, J. V. Dillon and K. Murphy, "Deep Variational Information Bottleneck", *Proceedings of the 5th International Conference on Learning Representations (ICLR),* Toulon, France, 2017.

[5] R. T. Richtmyer and J. A. Thomas, *Difference Methods for Initial-Value Problems,* 2nd ed.*,* New York, 1967.

[6] N. Tishby, F. C. Pereira and W. Bialek, "The Information Bottleneck Method", *arXiv:physics/0004057,* 2000.

[7] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Second Edition. Wiley, New York, 2006.

[8] D. P. Kingma and M. Welling, "An Introduction to Variational Autoencoders", *Foundations and Trends in machine Learning,* vol. 12, no. 4, pp. 307-392, 2019.

[9] Pytorch home page. [Online]. Available: https://pytorch.org/