PARAMETER-EFFICIENT VISUAL LEARNING WITH INFORMATION-PRESERVING TRANSFORMATIONS FOR DEEP CONVOLUTIONAL NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks typically transform representations through successive layers while discarding information that is considered unnecessary for the next transformation. Inspired by predictive coding, we propose the **Representation Error Residual (RER)**, a feedforward representation in which each layer explicitly separates its predictable component from the information that cannot be reconstructed by the layer's own transformation. Given an input representation, a convolutional pathway produces the primary feature representation while a tied reconstruction pathway estimates the input; their difference forms a local novelty signal. This signal is projected into the feature space and combined with the conventional representation, conserving information that is poorly represented by the current prediction for subsequent layers. We develop **RERNet**, a convolutional architecture built from novelty-preserving blocks, and investigate how the resulting representation evolves with depth. On CIFAR-100, an 8-block RERNet-L achieves 80.95% accuracy, comparable to a ResNet-18 (81.1%) with only 34% as many parameters, and approaches the performance of a ResNet-34 (83.0%) with only 18% as many parameters. We therefore present RER as both a representation mechanism and a means of trading parameter capacity for computation, providing a compact alternative to conventional residual representations.

1 INTRODUCTION

Modern visual recognition systems depend on deep convolutional neural networks (CNNs) that progressively transform an input into increasingly abstract representations. Although deeper and wider architectures can improve representational capacity, this improvement is often accompanied by substantial increases in parameter count, memory requirements, and the cost of moving model weights between memory and compute units. This creates a tension for resource-constrained systems: reducing the number of parameters can substantially reduce the memory footprint, but conventional approaches to parameter efficiency often achieve this by restricting the computational capacity of the network. A central question in the design of such networks is not only how to improve predictive performance, but also how information should be transformed and preserved as representations propagate through depth.

A particularly influential example of a simple architectural modification with substantial consequences is *residual networks* (ResNets) (He et al., 2016b). While a conventional convolutional block applies a learned transformation, a *residual block* instead introduces an explicit pathway for the original representation to which it adds a learned modification map. The development of residual networks is a breakthrough in deep learning research that demonstrably advanced the scalability of large deep learning vision models. A well-fit ResNet learns hidden spaces such that the model effectively predicts the difference between input and the expected output progressively layer-by-layer. The residual formulation allows information to persist through the network while the learned transformation focuses on modifying that representation. Residual connections have consequently become a fundamental component of modern vision architectures and are also found in models for other domains such as large language models (Vaswani et al., 2017).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In this work, we take this observation as a starting point and investigate whether the residual-network principle can be modified in a different direction. ResNets avoid information loss by forcing a linear component on to the transformation of each layer. An alternative perspective is to ask whether a representation can effectively isolate the information already present in each layer. Predictive coding provides a useful conceptual basis for this question. A long-standing idea in neuroscience is that sensory systems remove predictable information and preferentially represent surprising information. For example:

- retinal ganglion cells respond strongly to changes rather than absolute brightness (Freeman et al., 2010),
- auditory neurons respond strongly to unexpected tones (Xu et al., 2014),
- cortical neurons adapt to repeated stimuli (Grill-Spector et al., 2006; Xu et al., 2014).

Predictive-coding models distinguish between information that can be predicted from an existing representation and the residual information that remains unexplained. The prediction error therefore provides an explicit signal about aspects of the input that are not captured by the current prediction. While predictive coding has primarily been studied as a model of neural computation and representation, its underlying principle can also motivate feedforward representation-learning mechanisms. This leads to the central hypothesis of this work: *the representation error of a learned transformation can provide complementary information for subsequent layers*. We investigate whether this principle can be incorporated into the ResNet framework through a modification of the layer transformation.

Specifically, given an input x, we first transform it into a representation z, which is then used to construct a reconstruction $\hat{x}$ of the input. The difference defines a local reconstruction error, which we refer to as the *novelty* signal. We project this signal into the feature space and preserve it alongside the conventional feature representation. The resulting transformation can therefore be viewed as an experimental counterpart to the residual transformation: while a residual block explicitly preserves the input, our proposed block extracts the information that is not explained by the learned transformation.

We refer to this mechanism as a *Representation Error Residual* (RER) block and construct RERNet by stacking such blocks, as shown in Fig. 1. The reconstruction pathway uses the weights of the forward transformation in a tied transposed-convolution operation, giving the reconstruction error a direct relationship to the representation produced by the layer.

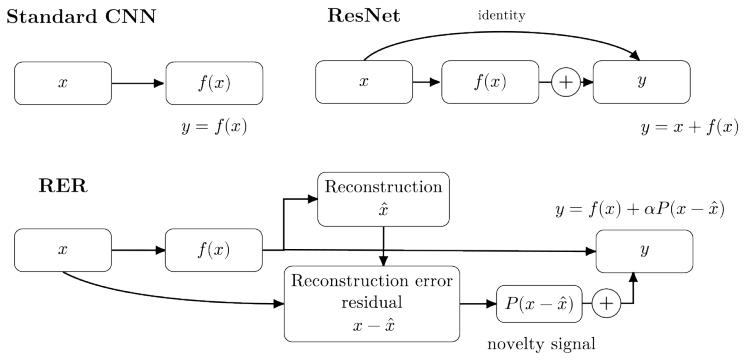


Figure 1: The RER block compared with standard CNN and ResNet blocks. While ResNets add an identity mapping to the output, RERNet blocks project a linear mapping of the reconstruction error into the conventional CNN output.

An additional consequence of this design is a different parameter–computation trade-off from conventional compact architectures. The reconstruction and novelty pathways introduce additional computation while allowing competitive representations to be constructed with substantially fewer learned parameters. This approach was taken with the motive to do more work per parameter and

make model operation compute-bound. Such a trade-off could be relevant to resource-constrained inference settings where parameter storage and weight movement can constitute significant costs.

We evaluate RERNet on CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) and compare it with conventional residual networks ResNet-18 and ResNet-34. We also compare performance with MobileNet-V2 (Sandler et al., 2018), a vision model architecture aimed at computational efficiency. In addition, we test the ability of the best-performing configuration to act as a backbone in a segmentation model on the Oxford-IIIT Pet Dataset (Parkhi et al., 2012). Beyond classification performance, we analyze the representations produced by the network through reconstruction and novelty visualizations, with the goal of characterizing the information captured by the proposed RER mechanism.

2 Related Work

2.1 Residual Networks and the Mechanics of Deep Representations

Residual networks were introduced as a simple reformulation of the transformation performed by a deep network. Rather than directly learning a convolutional mapping, a residual block learns a residual function relative to its input and outputs

$$H(x) = x + F(x), \tag{1}$$

where $F(x)$ is the learned convolutional map. The identity mapping provides a direct pathway for the input through the network (He et al., 2016b). This modification addressed an important limitation of conventional deep networks: increasing depth can make optimization increasingly difficult even when the deeper architecture should, in principle, be capable of representing the functions of a shallower network. It was shown that residual networks could be trained to substantially greater depths than conventional architectures, including networks with hundreds of layers on ImageNet and up to 1000 layers on CIFAR-10 (He et al., 2016b). In addition, He et al. (2016a) analyzed the propagation of signals through residual blocks and showed that identity skip connections provide particularly direct pathways for both forward activations and backward gradients. Their analysis motivated a pre-activation residual unit in which normalization and nonlinear transformations precede the addition of the identity pathway, allowing the skip connection itself to remain a clean identity mapping.

Another line of work has examined the effective computational structure of residual networks. Veit et al. (2016) proposed viewing a residual network as a collection of paths of different lengths created by the combination of residual and identity branches. Through layer-removal and reordering experiments, they showed that trained residual networks can tolerate certain structural modifications at inference time and argued that their behavior is consistent with an ensemble of relatively shallow paths. The scalability of residual architectures has also motivated mechanisms that explicitly manipulate their effective depth. For example, stochastic depth (Huang et al., 2016) randomly bypasses residual blocks during training while retaining the complete network at inference time, allowing extremely deep residual networks to be optimized more efficiently. The relation of the functional structure of residual networks and depth is thus distinct from typical feed-forward CNNs, with emergent implicit computational paths and residual scaling controlling signal magnitude (Dherin & Munn, 2025). Collectively, these studies demonstrate that the identity pathway introduced in residual learning significantly changes the dynamics of information propagation, optimization and the functional structure of trained networks.

The impact of residual learning extends beyond image classification tasks. The original ResNet models were used as components of object detection and segmentation systems, demonstrating that the learned residual representations could serve as general-purpose visual features (He et al., 2016b). Subsequent vision architectures have adopted residual networks as backbones for a wide range of tasks, including region-based object detection, semantic and instance segmentation, and multi-scale feature extraction. For example, Feature Pyramid Networks combine a bottom-up convolutional backbone with a top-down pathway and lateral connections to construct multi-scale representations, with residual networks providing a standard backbone configuration (Lin et al., 2017). Similarly, Mask R-CNN extends region-based detection to instance segmentation using a ResNet-FPN backbone (He et al., 2017).

2.2 Predictive Coding and Reconstruction Error Representations

Predictive coding provides the second conceptual foundation for the proposed RERNet architecture. In its classical formulation, predictive coding describes perception as a hierarchical process in which higher-level representations generate predictions of lower-level sensory activity, while discrepancies between predictions and sensory observations are propagated as prediction errors. Rao & Ballard (1999) introduced an influential computational model of predictive coding in the visual cortex in which feedback connections carry predictions of lower-level activity and feedforward connections carry residual errors between predictions and observed activity. They showed that models equipped with such a predictive hierarchy could account for several properties of visual receptive fields.

The broader predictive-coding literature has developed this principle into a family of models with differing assumptions about the generative model, inference procedure, and representation of prediction errors. Spratling (2017) reviews several predictive-coding algorithms, describing how the term encompasses multiple computational formulations rather than a single canonical algorithm. Predictive coding has been connected to Bayesian inference, variational formulations, and learning algorithms for artificial neural networks (Friston, 2005; Buckley et al., 2017; Millidge et al., 2024). These developments have motivated predictive-coding-inspired neural networks in which prediction errors play an explicit role in inference and learning.

An important distinction is therefore necessary between predictive coding as a theory of neural computation and the mechanism proposed here. RERNet does not implement iterative inference, recurrent prediction-error minimization, or a complete cortical predictive-coding circuit. It simply adopts a single computational motif from predictive coding: the explicit construction and propagation of the discrepancy between an *input representation* and a *coarse reconstruction from deeper internal representations* of that input. The reconstruction error is subsequently treated as a feature representation; it is not treated as an optimization signal. This distinction allows us to investigate the representational value of the errors independently of the broader assumptions inherent to predictive-coding models.

2.3 Position of the Present Work

The present work lies at the intersection of these two lines of research. ResNets demonstrate that a simple modification to the information pathway of a conventional CNN can substantially alter its optimization behavior, scalability, and effective computational structure. Predictive coding provides a complementary principle in which prediction errors represent information that is not explained by a current model. We combine these perspectives by replacing the conventional identity-preserving intuition with an explicit reconstruction error pathway. Hence, we study the utility of reconstruction error as a component of learned visual representations. This formulation also leads naturally to a parameter–computation trade-off, which we investigate alongside representation quality and classification performance.

3 Methodology

3.1 Overview

Our objective is to investigate whether information that is not explained by a learned feature transformation can be retained as a useful representation for subsequent layers. We formulate this as a modification of the residual learning principle. A conventional convolutional block response maps an input tensor X to an embedding Z using an activation function σ, which can be formulated as

$$Z = \sigma(W * X), \tag{2}$$

where residual networks augment the mapping as

$$Z = X + \sigma(W * X). \tag{3}$$

On minimizing the discrepancy between Z and a target output O, the underlying neural network is driven to approximate the residual $O - X$. In contrast, our proposed block constructs a coarse estimate $\hat{X}$ of the input from Z and explicitly projects the resulting reconstruction residual $\hat{X} - X$

into Z. The resulting Representation Error Residual (RER) block consists of four operations: feature extraction, input reconstruction, residual computation, and projection of the residual into the feature space. The resulting representation is given by

$$z = \phi(\mathrm{BN_z}(W * X)), \tag{4}$$

where BN_z is batch normalization and ϕ is a nonlinear activation function. A reconstruction $\hat{x}$ is then obtained using a weight-tied transposed convolution,

$$\hat{x} = \mathrm{ConvTranspose2D}(z, W), \tag{5}$$

and the local residual is defined as

$$e = x - \hat{x}. \tag{6}$$

The residual is projected into the feature space using a learned mapping P, referred to as the *novelty signal*, whose contribution is modulated by a layer-level parameter α.

$$y = z + \alpha P(e). \tag{7}$$

This formulation allows the network to interpolate between a conventional convolutional representation and one in which feature information is explicitly preserved.

3.2 Weight-Tied Coarse Reconstruction

The reconstruction pathway is constructed as a transpose convolution from the weights of the forward convolution rather than introducing an independent decoder. The stride and padding are selected to approximately invert the spatial transformation of the forward convolution. Weight tying has two purposes. First, it reduces the number of additional trainable parameters required by the reconstruction pathway. An independently parameterized reconstruction network would substantially increase the parameter footprint and make it difficult to distinguish the effect of preserving reconstruction errors from simply increasing model capacity. Second, the tied reconstruction directly relates the error residual to the compression error of the transformation learned by the convolutional layer. The resulting error therefore measures the component of the input that is not captured by the current feature mapping rather than a mapping produced by an independently trained auxiliary network. When a convolution changes spatial resolution, the reconstructed representation is spatially aligned with the original input before the error is computed. This permits the reconstruction error residual to be incorporated into blocks that perform down-sampling.

3.3 Projecting the Representation Error Residual

The representation error e initially lies in the input feature space and therefore generally has a different number of channels from the transformed representation z. We introduce a learned 1×1 convolution to map it into the output feature space:

$$P(e) = BN_e(W_P * e), \tag{8}$$

where W_P contains the parameters of the novelty projection and BN_e is an independent batch normalization for the novelty stream. This allows the statistics of the novelty stream to be determined independently of the statistics of the learned feature representation. This projection operates across channels at each spatial location and does not perform additional spatial filtering. Consequently, the spatial structure of the novelty signal is preserved while its channel representation is adapted to the feature space of the current layer. The two streams are subsequently combined according to Eq. 7.

3.4 RESIDUAL RER BLOCKS

Although the novelty signal provides an alternative to the identity pathway of a conventional residual block, we additionally investigate a formulation in which the RER transformation is embedded within an overall residual block. In this formulation, the RER mechanism operates internally while the block retains an explicit shortcut connection:

$$y = z + \alpha P(x - \hat{x}) + S(x), \tag{9}$$

where $S(x)$ denotes the conventional residual shortcut. When the input and output dimensions are identical, S is the identity mapping. When the spatial resolution or number of channels changes, S is implemented using a 1×1 convolution with the corresponding stride.

The internal RER mechanism determines what information is not explained by the learned transformation, while the outer residual pathway provides a direct route for the input representation through the block. This allows us to investigate whether RER representations provide useful information in addition to the established optimization and information-preservation properties of residual connections.

3.5 NOVELTY SIGNAL STRENGTH

The contribution of the novelty stream is controlled by the scalar α. Rather than treating this parameter as an independently learned weight, we use a depth-dependent schedule. For a network containing N RER blocks, the novelty strength of block i is

$$\alpha_i = \alpha_{\max} \cos\left(\frac{\pi i}{2(N-1)}\right), \qquad i = 0, \ldots, N-1. \tag{10}$$

Thus, early blocks receive a larger contribution from the novelty signal, while the contribution decreases with depth. This provides a gradual transition from strongly biased RER blocks toward conventional residual blocks. The final block has $\alpha \approx 0$, and is consequently implemented as a conventional residual block that avoids unnecessarily computing the RER mechanism when its contribution is negligible. The depth-dependent schedule is motivated by the different roles of shallow and deep representations. Early visual layers operate on relatively low-level spatial structure, where the distinction between predictable and unexplained local information can be directly associated with image structure. Deeper layers instead encode increasingly abstract features, so we reduce the contribution of the representation error.

3.6 RERNET ARCHITECTURE

RERNet is constructed by stacking RER blocks following a convolutional image-classification architecture. The network begins with a 3×3 convolutional stem followed by batch normalization and a ReLU activation. The resulting representation is processed by a sequence of RER blocks, followed by global average pooling and a linear classification layer.

We consider three width configurations:

- **RERNet-S**: all RER blocks use 64 output channels.
- **RERNet-M**: the first block uses 64 channels and all subsequent blocks use 128 channels.
- **RERNet-L**: the first block uses 64 channels, the second uses 128 channels, and subsequent blocks use 256 channels.

Spatial downsampling is performed at intermediate stages of the network using stride-2 RER blocks. The corresponding shortcut projection ensures that the dimensions of the residual pathway match those of the transformed representation. The number of RER blocks is deliberately smaller than the number of convolutional blocks in the ResNet baselines. As ResNet-34, which uses 16 residual blocks, is the largest standard model we test against, we limit our experiments to models with half as many RER blocks to ensure proportional computational budget as the RER mechanism includes more operations. This design allows us to investigate whether the RER mechanism can achieve

Model	Parameters	Accuracy (CIFAR-100)	Accuracy (CIFAR-10)
ResNet-34	21,328,292	83.05	97.27
ResNet-18	11,229,752	81.10	96.63
RERNet-L8	3,780,426	80.95	96.60
RERNet-M8	1,138,826	77.12	95.67
RERNet-S8	336,906	69.96	93.51
RERNet-L4	1,089,290	72.2	92.35
RERNet-M4	464,266	69.18	91.49
RERNet-S4	171,786	62.23	89.44
MobileNet-V2	5,194,748	75.72	94.46

Table 1: Results for top-1 classification accuracy on CIFAR-100 and CIFAR-10 datasets, comparing RERNet variants with ResNet and MobileNet-V2 models. Parameter counts listed are for the CIFAR-100 variant. The number following the RERNet architecture states the number of blocks.

Backbone	IoU	Dice score
ResNet-34	0.8909	0.9421
ResNet-18	0.8906	0.9420
RERNet-L8	0.8677	0.9289

Table 2: Segmentation results on the Oxford IIIT Pet dataset using a feature pyramid network decoder on an 8-block RERNet-L and ResNet models.

comparable representation quality with a substantially smaller depth and parameter count, while having an arithmetically intensive computation.

4 Results and Discussion

4.1 Performance

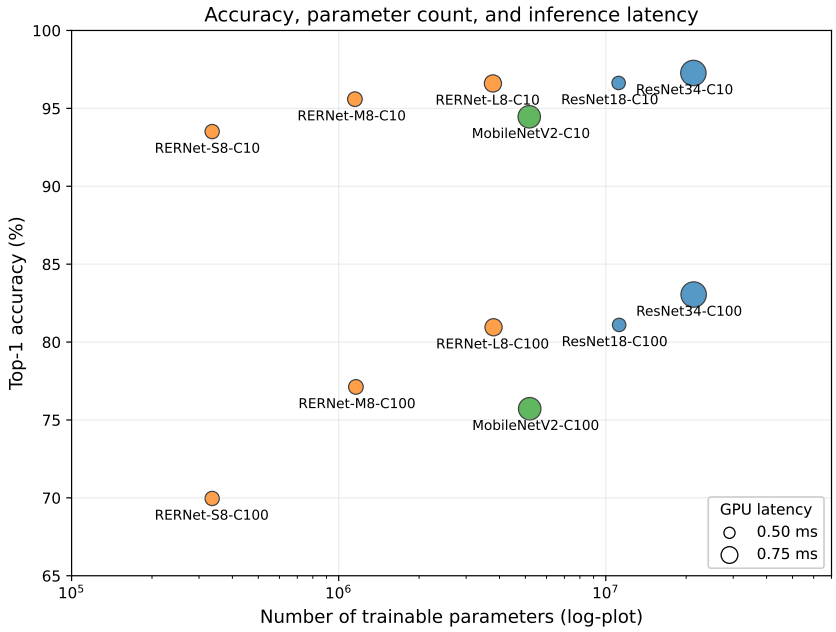


Figure 2: Semilog plot for parameter count against accuracy of tested models, where radii are linearly proportional to single-input GPU latency in milliseconds on an NVIDIA GeForce RTX 5090.

Model	MACs/image	FLOPs/image	Latency (ms)
ResNet-34	1,161,504,768	2,323,009,536	1.1379
ResNet-18	556,705,792	1,113,411,584	0.6008
RERNet-L8	1,297,556,480	2,595,112,960	0.7701
RERNet-M8	561,385,984	1,122,771,968	0.6562
MobileNet-V2	25,671,936	51,343,872	1.0057

Table 3: Computational costs in MAC count, FLOPs, and mean GPU inference latency. All experiments with batch size 1 on an NVIDIA GeForce RTX 5090.

Table 1 lists the classification accuracies of various RERNet variants on the CIFAR-10 and CIFAR-100 datasets, compared with ResNet-34, ResNet-18, and MobileNet-V2 along with the respective parameter counts. An 8-block RERNet-L achieves 80.95% classification accuracy, compared with 81.1% for ResNet-18 and 83.0% for ResNet-34, while using 34% and 18% the parameters of ResNet-18 and ResNet-34 respectively. We also compare the best performing RERNet variant with ResNet-34 and ResNet-18 on their utility as encoders for segmentation in Table 2. We use the Oxford IIIT Pet dataset (Parkhi et al., 2012) to test the ability of the models to act as backbones extracting useful visual features at different depths. For each experiment, feature maps at 4 levels of depth were extracted from the models using a feature pyramidal network decoder of approximately 850k parameters. As summarised in Fig. 2, the RERNet architecture demonstrates competitive representational capacity and throughput on visual tasks using a parameter-efficient network. Training details are listed in Sec. A.1 of the appendix.

4.2 Interpreting the Novelty Maps

We hypothesize that novelty projection maps act as a coarse salience adjustment mechanism, modulating the representation of spatial regions containing information that is poorly captured by the current layer but remains useful for subsequent processing. Conventional convolutional filters learn feature detectors whose responses are optimized to support the downstream task. In an RER block, the projected reconstruction error introduces features back into the main pathway. During training, the 1×1 novelty projections learn to enhance those features which tend to be lossy for prediction-specific information. As we progress deeper in the model, the RER mechanism is expected to learn a channel-specific response modulation. If the mechanism is working correctly then across all channels, we should observe saliency-aware activation, otherwise the 1×1 projection maps should be either random or silenced by training.

In Fig. 3, we visualize novelty projection maps of an 8-block RERNet-L spatially for CIFAR-100 samples to determine whether the novelty pathway captures structured information rather than behaving as unstructured noise. The RER mechanism pathway has channels equal in number to the output of the standard convolutional pathway; the figure aggregates the signal response across channels. We observe that the aggregate response is closely tied to class-specific regions and hence helping the model identify regions concomitant for prediction.

4.3 Computational Cost

The reconstruction operation does not introduce independently learned parameters as its weights are tied to the forward convolution. The additional parameter cost of the RER mechanism is limited to the novelty projection map, substantially smaller than that of introducing an independent reconstruction network. However, reconstruction and projection introduce additional computation. RERNet therefore intentionally trades computation for parameter efficiency. We therefore characterize models using computation measured in MACs and FLOPs per image, and also measure mean GPU inference latency, listed in Table 3. Computation statistics were calculated using the `fvcore` library [1]. The counts are slightly underestimated due to some operations not being compatible with the performance analysis, but we strongly believe that the statistics are representative of the relative compute costs. All experiments were performed using an NVIDIA GeForce RTX 5090 on Python 3.13.5, PyTorch 2.9.0.dev and `torchvision` 0.24.0.dev.

[1] https://github.com/facebookresearch/fvcore

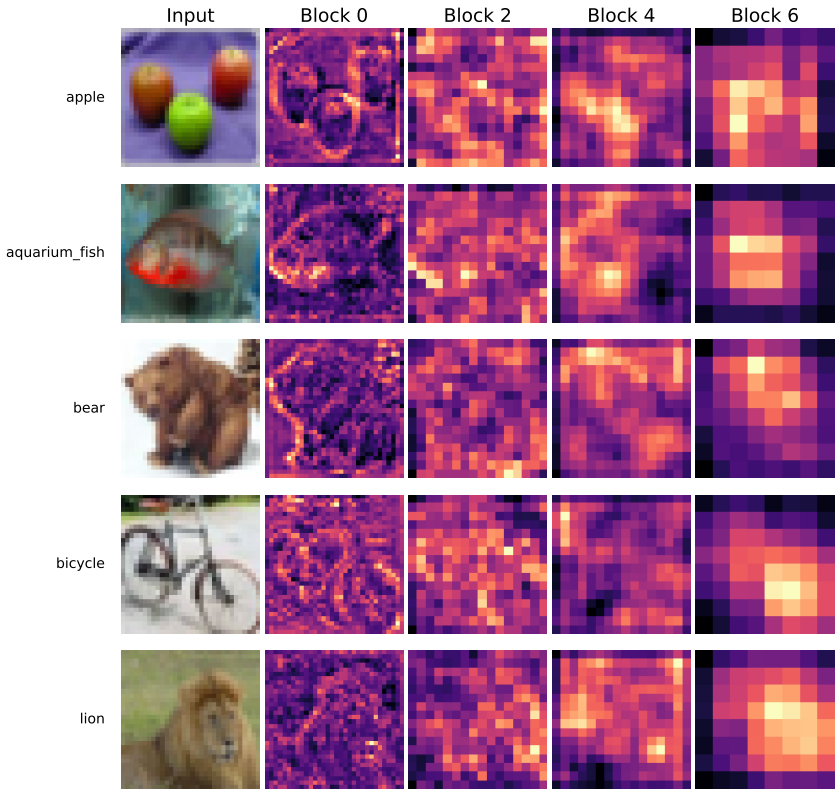


Figure 3: Visualizing novelty signal maps of different CIFAR-100 samples at different network depths aggregated across output channels of an RERNet-L8 trained on the dataset. The 1×1 projection maps progressively become correlated with class-specific saliency deeper into the model.

We observe a higher MAC and GFLOP count for RERNets, however they have lower mean GPU inference latency, which suggests that it is computationally beneficial to push CNN models further into compute-bound regimes for better hardware utilization. In comparison, although MobileNet-V2 has substantially lower computational costs, its inference latency is higher due to low parallelism of depthwise convolutions and overheads of managing memory buffers and kernel launches for implementing inverted residuals.

4.4 Conclusion and Future Work

We introduced a predictive-coding-inspired residual representation in which reconstruction errors are projected back into the feature space and incorporated alongside conventional convolutional representations. Despite using substantially fewer parameters than the ResNet baselines considered, the proposed architecture achieves competitive classification performance, while the visualization of novelty maps suggests that the reconstruction pathway can capture spatially localized information that is complementary to the primary feature representation. Our results provide initial evidence that RERNets can serve as an alternative to conventional ResNets, while pushing CNN models more into compute-bound regimes that better utilize GPU hardware.

It is not yet clear whether the depth scalability observed in standard ResNets is preserved by the proposed architecture. The interaction between reconstruction errors, feature transformations, and depth requires further investigation. Further experiments are needed to examine depth scaling, transfer to additional visual tasks, and quantitative measures of the information and salience captured by the novelty representation. Future work also includes alternatives to the current reconstruction pathway that mimic the representational benefits of the representation error residual.

AI USE STATEMENT

We have not used generative AI tools for any tasks requiring disclosure under the ICLR 2027 AI Policy like formatting, language editing, theoretical formulation, development of proposed methods, experimental design, data analysis, interpretation of results or generation of research claims. We take responsibility for the final content of this work, including text, claims or artifacts.

REPRODUCIBILITY STATEMENT

We provide the complete source code used for all experiments as supplementary material, including the implementation of the proposed method, baseline methods, training and evaluation scripts, configuration files, and scripts for generating the reported results. Environment and dependency specifications are also provided. The main paper and appendix describe the model, training procedure, experimental settings, and ablations needed to understand and reproduce the results. All experiments reported in the paper can be reproduced using the provided code and configurations.

REFERENCES

Christopher L. Buckley, Chang Sub Kim, Simon McGregor, and Anil K. Seth. The free energy principle for action and perception: A mathematical review. *Journal of Mathematical Psychology*, 81:55–79, 2017. ISSN 0022-2496. doi: https://doi.org/10.1016/j.jmp.2017.09.004. URL https://www.sciencedirect.com/science/article/pii/S0022249617300962.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Benoit Dherin and Michael Munn. On residual network depth. *arXiv e-prints*, art. arXiv:2510.03470, October 2025. doi: 10.48550/arXiv.2510.03470.

Daniel K. Freeman, Gilberto Graña, and Christopher L. Passaglia. Retinal ganglion cell adaptation to small luminance fluctuations. *Journal of Neurophysiology*, 104(2):704–712, 2010. doi: 10.1152/jn.00767.2009. URL https://doi.org/10.1152/jn.00767.2009. PMID: 20538771.

Karl Friston. A theory of cortical responses. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 360(1456):815–36, 2005. doi: 10.1098/rstb.2005.1622.

Kalanit Grill-Spector, Richard Henson, and Alex Martin. Repetition and the brain: neural models of stimulus-specific effects. *Trends in Cognitive Sciences*, 10(1):14–23, Jan 2006. ISSN 1364-6613. doi: 10.1016/j.tics.2005.11.006. URL https://doi.org/10.1016/j.tics.2005.11.006.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 630–645, Cham, 2016a. Springer International Publishing. ISBN 978-3-319-46493-0.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016b.

Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep networks with stochastic depth. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (eds.), *Computer Vision – ECCV 2016*, pp. 646–661, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46493-0.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, 2017. doi: 10.1109/CVPR.2017.106.

Beren Millidge, Mufeng Tang, Mahyar Osanlouy, Nicol S. Harper, and Rafal Bogacz. Predictive coding networks for temporal prediction. *PLOS Computational Biology*, 20(4):1–31, 04 2024. doi: 10.1371/journal.pcbi.1011183. URL https://doi.org/10.1371/journal.pcbi.1011183.

Samuel G. Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 774–782, October 2021.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Rajesh P. N. Rao and Dana H. Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature Neuroscience*, 2(1):79–87, Jan 1999. ISSN 1546-1726. doi: 10.1038/4580. URL https://doi.org/10.1038/4580.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, 2018. doi: 10.1109/CVPR.2018.00474.

M.W. Spratling. A review of predictive coding algorithms. *Brain and Cognition*, 112:92–97, 2017. ISSN 0278-2626. doi: https://doi.org/10.1016/j.bandc.2015.11.003. URL https://www.sciencedirect.com/science/article/pii/S027826261530035X. Perspectives on Human Probabilistic Inferences and the 'Bayesian Brain'.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/37bc2f75bf1bcfe8450a1a41c200364c-Paper.pdf.

Xinxiu Xu, Xiongjie Yu, Jufang He, and Israel Nelken. Across-ear stimulus-specific adaptation in the auditory cortex. *Frontiers in Neural Circuits*, Volume 8 - 2014, 2014. ISSN 1662-5110. doi: 10.3389/fncir.2014.00089. URL https://www.frontiersin.org/journals/neural-circuits/articles/10.3389/fncir.2014.00089.

A Appendix

A.1 Training Pipeline

All experiments were performed for 300 epochs with a batch size of 128. We used an SGD optimizer with a peak learning rate of 0.1 linearly warmed up over the first five epochs and a weight decay parameter 5×10^{-4}. Cross entropy loss with a label smoothing of 0.1 was used for both classification and segmentation. The ResNet and RERNet models were trained with the following augmentation routine for the training dataset:

- `transforms.RandomCrop(32, padding=4)`
- `transforms.RandomHorizontalFlip()`
- `transforms.TrivialAugmentWide()` (Müller & Hutter, 2021)

- `transforms.Normalize(Dataset_Mean, Dataset_STD)`
- `Cutout(N_holes=1, length=16)` (DeVries & Taylor, 2017)

MobileNet-V2 fails to scale to CIFAR-100 with this pipeline achieving less than 70% top-1 accuracy. Its training pipeline replaced `TrivialAugmentWide` with a subpolicy which would randomly select from shear, translate, rotate, color balance, posterize, solarize, contrast, sharpness, brightness, autocontrast, equalize and invert transforms using fixed random choice probabilities.

A.2 Implicit Scaling of Novelty Signal with Batch Norm

Naïvely adding the novelty signal to the convolutional transform would require the novelty map weights to learn the correct scale of the signal. If the RER map is initialized incorrectly or fails to scale over training, the credit assignment happens to the convolutional map in tandem so there is poor division of learning between the two mechanisms. A poorly learnt novelty map would either be too weak to impart information or conversely be too large and override important features of the output. The model thus needs to be able to learn the scale of both branches reliably. As one block passes backpropagation gradients to its predecessor block, it is not obvious how credit assignment for errors is to be distributed between the convolutional parameters and the novelty pathway. For this reason, as is seen in Eq. 4, we intentionally introduce a batch normalization step BN_z between the convolution and transpose convolution steps such that the reconstruction does not correct for the normalization. This imparts a scale to the input of the novelty map; its output can then be scaled using its own normalization step BN_e, as shown in Eq. 8. These magnitudes can then be learnt by the model adequately by adjusting the scale parameters of BN_z and BN_e independently during training.