# Scientific Background

Deep learning has experienced remarkable progress over recent years, resulting in its extensive integration across diverse domains. However, amidst these significant advancements, the field faces fundamental challenges that demand our immediate attention. Once these aspects are successfully addressed, we can anticipate a significant enhancement in the performance of deep learning.

The first challenge is *explainability*. Despite deep learning's exceptional practical achievements, the reasons for its amazing success remain shrouded in mystery, and the quest for explainability aims to shed light on why deep learning yields such remarkable results.

The second challenge, which is the focus of this research proposal, is *generalization*. Generalization, a cornerstone of deep learning, embodies a model's capacity to seamlessly extend its performance beyond the boundaries of its training data. It signifies the model's agility in swiftly and accurately adapting to new, unseen data points. Generalization is highly dependent on the deep model complexity. For example, it is a significant issue especially when dealing with over-parametrized learning architectures, where the number of parameters is significantly larger than the amount of training data [20, 30 – “Exploring Generalization in Deep Learning”]. In such an over-parametrized setting, the objective has multiple global minima, all minimize the training error, but many of them do not generalize well. Hence, just minimizing the training error is not sufficient for learning: picking the wrong global minima can lead to overfitting to the training data and to poor generalization behavior. In such scenarios, generalization behavior depends implicitly on the algorithm used to minimize the training error.

## Deep Learning Generalization

Deep learning generalization aims to learn generic feature representations agnostic to domains (or datasets) and make trained models perform well in completely new domains. Toachieve this challenging goal, one needs to train models that can capture usefulinformation observed commonly in multiple domains and recognize semanticallyrelated but visually inconsistent examples effectively. Many real-world problemshave similar objectives so this task can be widely used in various practical applications.

Generalization techniques are classified into several groups depending on their approaches. Some algorithms define novel loss functions to learn domain-agnostic representations [7,15,20,21] while others are more interested in designing deep neural network architectures to achieve similar goals [6,13,17]. The proposed research belongs to the second category, i.e. network architecture design methods. In particular, we will explore and introduce novel adaptive models for the three following key-components of deep learning: *data normalization*, *regularization* and *optimization* (“Learning to Optimize Domain Specific Normalization for Domain Generalization”). **These adaptive models enhance both performance and stability, striking a delicate balance between model simplicity and complexity.**

**In other words, our adaptive models, as proposed, empower us to achieve the same task accuracy for simple models as is obtained by complex models while mitigating its risk of overfitting that is common for complex models but not for the simple ones.**

## Data Normalization

Data normalization and deep model generalization are closely related in the context of training deep neural networks. Data normalization is a preprocessing technique used to scale and center the input data, while deep model generalization refers to a model's ability to perform well on unseen data. Normalization affects model generalization due to the following:

1. **Stabilizes Training**: Deep learning models, particularly neural networks with many layers, are sensitive to the scale and distribution of input features. When the input features have vastly different scales, some weights in the network may update much more slowly than others during training. This can lead to convergence issues and longer training times. Data normalization scales all input features to a similar range, which helps in stabilizing the training process.
2. **Faster Convergence**: Normalizing the data often results in faster convergence during training. This is because the optimization algorithm can more effectively navigate the loss landscape when the features are on a similar scale. Faster convergence can lead to better generalization since the model doesn't spend as much time fitting the training data noise.
3. **Mitigates Overfitting:** Data normalization can also help in preventing overfitting. Overfitting occurs when a model becomes too complex and starts fitting the noise in the training data rather than capturing the underlying patterns. When input features are normalized, it becomes less likely for the model to overfit because it focuses on the relevant patterns rather than the noise.
4. **Improved Gradient Flow:** Normalizing the data can lead to improved gradient flow during backpropagation. This means that the gradients used to update the model's weights are less likely to explode or vanish. Improved gradient flow can help the model generalize better because it can learn more effectively from the training data.
5. **Transferability**: When you train a deep model on normalized data, it's more likely to transfer well to new and unseen datasets. Since the model has learned patterns that are invariant to the scale and distribution of the input features, it's better equipped to handle a broader range of data, improving its generalization capabilities.

In summary, data normalization plays a crucial role in enhancing the generalization capabilities of deep learning models. It helps stabilize training, prevent overfitting, and ensures that the model can effectively learn and generalize patterns from the data. A variety of established normalization techniques have been devised to address these challenges. For instance, Batch Normalization [16] conducts global normalization across the batch dimension. However, its sensitivity to batch size and the impracticality of batch-wise normalization during inference necessitate the use of pre-computed mean and variance values derived from the training set via running averages [16]. Nevertheless, these pre-computed statistics may become unreliable when the distribution of the target data shifts, creating disparities between the training and testing phases. To mitigate issues related to the batch dimension, alternative normalization methods [17-19] have surfaced. Layer Normalization (LN) [17] operates over all channels along the layers dimension, while Instance Normalization (IN) [18] conducts calculations akin to BN on individual samples. Weight Normalization (WN) [19] introduces filter weight normalization as an approach. Despite their merits, these techniques sometimes fall short of achieving the accuracy levels demonstrated by Batch Normalization (BN) in various visual recognition tasks. A more recent technique, Group-Normalization (GN), addresses the batch dimension challenge by segmenting channels into groups and normalizing features within each group [20]. GN initializes channel groups based on sequential order in the first epoch and maintains the same groups throughout the whole training. However, this assumption relies on the premise that the deep network will progressively learn optimal channel weights, which is not universally valid. The effectiveness of this method exhibits variability. Despite all these normalization techniques, the intricate challenges posed by medical images persist as an ongoing pursuit. The impact of normalization on learning generalization is clear and well-known. Proper normalization techniques can contribute to improved generalization by making the training process more stable, reducing overfitting, and enhancing the model's ability to handle variations in the data distribution.

**In this research grant, our aim is to delve deeper into the normalization-generalization relationship, studying the impact of hybrid normalization on the generalization of deep network. To do that, we will introduce an innovative adaptive strategy that overcomes the constraints of existing normalization methods.**

## Exponential Moving Average (EMA) Optimization

Deep learning architectures heavily rely on adaptive and non-adaptive optimization algorithms. These methods improve convergence towards global minimum through techniques such as momentum, individual learning rates, and exponential moving averages. Example optimizers are SGD, RMSprop, Adam, AdaGrad, AdamW, AdaHessian and MAS that combines SGD and Adam for better convergence and generalization. Deep *EMA*-based (Exponential Moving Average) optimizers play a crucial role in enhancing the generalization capabilities of deep neural network models. The relationship between model generalization and Deep EMA-based optimizers can be explained as follows:

1. **Stabilizing Training**: Deep EMA-based optimizers introduce stability into the training process. They maintain a smoothed version of the model's weights over time, which can help the model avoid rapid fluctuations during training. This stabilization can prevent the model from fitting the noise in the training data (and identifying the real data patterns, as should be), thus lead to better generalization.
2. **Noise Reduction**: During the training of deep neural networks, the optimization process can sometimes encounter noisy updates, especially when using small batch sizes or when the loss landscape is complex. Deep EMA-based optimizers, by averaging parameter values over time, reduce the impact of such noise. This leads to more consistent weight updates and can help the model generalize better by focusing on the underlying patterns in the data.
3. **Regularization**: Deep EMA-based optimizers effectively act as a form of regularization. By maintaining an exponential moving average of the model's weights, they encourage the model to explore more stable regions of the parameter space, which often correspond to better generalization. This regularization effect can help prevent overfitting, where the model becomes too specialized to the training data.
4. **Memory of Past Information**: Deep EMA-based optimizers retain a memory of past parameter values. This memory is beneficial for generalization because it allows the model to remember useful information from earlier stages of training, even as it continues to adapt to the current data. This can be especially valuable when the dataset is large or noisy.
5. **Improved Exploration-Exploitation Balance**: Deep EMA-based optimizers help strike a better balance between exploration (trying out new weight configurations) and exploitation (refining the current best-known configuration). This balance can contribute to improved generalization by ensuring that the model explores a wider range of solutions before settling on a final one.

These benefits make Deep EMA-based optimizers a valuable tool for improving the ability of deep neural networks to generalize well to unseen data.

### Identifying trends (Refs should be taken from our paper)

Despite the extensive efforts made to bridge the generalization gap between stochastic gradient descent (SGD) variants and adaptive methods, the disparity persists \cite{gupta2021adam}. A key limitation of existing adaptive methods arises from their limited ability to adapt effectively and quickly to data trends during the optimization process. Maiya et al. introduced the Tom optimizer (Trend over Momentum) for Computer Vision tasks, which employs Holt's Linear Trend Model as a time series model to predict gradient trends \cite{Tom}. By leveraging the gradient rate of change between successive time steps, Tom introduces a trend component to enhance convergence. However, a drawback of Tom lies in its assumption of persistent seasonal gradient trends, which may not hold true in real-world scenarios. Notably, in the domain of finance, Kolkova et al. demonstrated that technical indicators typically outperform Holt's smoothing in identifying trends in seasonality-free financial data \cite{kolkova2018indicators}. To address this limitation, our proposed approach aims to enhance adaptive methods by integrating a powerful technical indicator into the optimization process. By doing so, adaptive methods become more responsive to trend changes during the optimization process, leading to improved performance and better adaptation to real-world data dynamics.

### Exponential Moving Average (EMA)

*EMA* is a widely used technical indicator that applies exponentially decreasing weighting factors to past data, smoothing out short-term fluctuations, thus it is utilized to denoise the data during the optimization process. Denoting a data sequence up to time , *EMA* combines recursively the current and previous data values by

 is a tuning parameter in EMA that determines its responsiveness to data changes, where lower values provide faster response but less effective denoising. Double Exponential Moving Average (DEMA) and Triple Exponential Moving Average (TEMA) are extensions of EMA used in finance for trend assessment (\cite{Mulloy}). They incorporate lag correcting terms to enhance noise reduction and trend identification. TEMA demonstrates superior responsiveness to rapid data changes compared to EMA. The inclusion of lag correcting terms in TEMA reduces lag while maintaining smoothness. It is well known that the choice of deep optimizer can affect the network performance.

## Learning Regularization

The dependence between regularization and deep model generalization is a fundamental concept in machine learning and deep learning. Regularization techniques are employed to improve a model's generalization performance, which refers to its ability to make accurate predictions or classifications on unseen data. It is done thanks to the following:

1. **Prevention of Overfitting**: Regularization techniques, such as L1 and L2 regularization (also known as weight decay), dropout, and early stopping, are designed to prevent overfitting. Overfitting occurs when a model becomes too complex and starts fitting the noise in the training data rather than capturing the underlying patterns. Regularization introduces constraints or penalties on the model's parameters, discouraging it from becoming overly complex. By doing so, regularization helps the model generalize better because it focuses on the relevant patterns rather than the noise.
2. **Smoother Decision Boundaries**: Regularization methods encourage the model to learn smoother decision boundaries. Instead of fitting the training data points exactly, the model learns to make predictions based on broader patterns and trends in the data. This results in more stable and generalizable predictions when applied to new, unseen data.
3. **Weight Pruning**: Techniques like L1 regularization encourage sparsity in model parameters by driving some weights to become exactly zero. This process effectively prunes unnecessary connections in the model, reducing its capacity to memorize training data and making it more likely to generalize well.
4. **Dropout as a Form of Noise Injection**: Dropout is a regularization technique that randomly deactivates a fraction of neurons during training. This introduces noise and variability into the training process, making the model more robust and less prone to overfitting. Dropout has been shown to improve generalization by preventing the model from relying too heavily on any single neuron or feature.

By striking a balance between fitting the training data and capturing the underlying patterns, regularization helps deep models perform well on new, unseen data, making them more robust and reliable in practical applications. To improve dropout performance, researchers have explored two main avenues:

* Sampling from Different Distributions: Some approaches involve sampling dropout masks from distributions other than Binomial (Ref).
* Adaptive Dropout Probabilities: Others focus on adapting the dropout probability based on prior knowledge of the network. For instance, Keshari, Singh, and Vatsa (2019) use stochastic gradient descent (SGD) to learn a strength parameter for guiding dropout regularization of each node. Meanwhile, Wang, Zhou, and Bilmes (2019) normalize the dropout probability at each layer and training batch to maintain a consistent effective dropping rate.

Most existing dropout regularization methods remove individual activations within each unit independently, either with a fixed or adaptive probability. To apply adaptive dropout it should be clear that visual structures in input images activate corresponding regions in convolution feature maps (He et al. 2015). This suggests that feature maps with similar activation patterns should be stochastically dropped to reduce co-adaptations. Nevertheless, these feature maps also encode information about intra-class variation in latent semantic features (Kim et al. 2017). This intriguing insight leads to explore the adaptive dropout approaches. The authors in [Group-Wise Dynamic Dropout Based on Latent Semantic Variations] propose 'group-wise dropout', a method that adapts to latent semantic variations while simulating dynamic sparseness in the network, ultimately improving object recognition performance. The authors in (Tompson et al. 2015) proposed spatial dropout, a random subset of activations in feature maps are dropped independently to reduce spatial correlations. Poernomo and Kang (2018) introduced 'cross-map dropout', which simultaneously drops or retains elements at the same coordinate on different feature maps. Additionally, Zhang, Yang, and Feng (2018) developed 'region dropout' by considering salient regions with fixed size and relative positions for training. However, the availability of these salient regions may not always be guaranteed for general object recognition problems. In the realm of adaptive dropouts, Wang and Manning (2013) demonstrated a Gaussian approximation to dropout, while Kingma, Salimans, and Welling (2015) proposed 'variational dropout' by connecting global uncertainty with dropout rates to optimize a generalized Gaussian dropout. Wager, Wang, and Liang (2013) analyzed dropout training as a form of adaptive regularization, while Ba and Frey (2013) updated the elementwise probability for mask matrix generation based on activation output. Zhuo, Zhu, and Zhang (2015) extended the overlaid model to learn adaptive dropout rates for different neurons or groups of neurons. Keshari, Singh, and Vatsa (2019) introduced 'guided dropout' to drop network nodes with high strength to encourage low-strength nodes, while Wang, Zhou, and Bilmes (2019) proposed 'Jumpout', which samples the dropout probability from a monotone decreasing distribution. These methods drop unit activations independently, but feature visualization studies have shown that interactive information between feature nodes can be beneficial for improving object recognition performance (Kim et al. 2017; Du et al. 2018). [Group-Wise Dynamic Dropout Based on Latent Semantic Variations]. Nonetheless, the primary hurdle faced by these various methodologies stems from their substantial computational demands and the departure from randomness. The papers on adaptive knowledge-based dropout make concerted efforts to thwart the co-adaptations between neurons through various strategies. However, if these efforts fall short, there exists a significant chance of overfitting.

# Research objectives & expected significance

**This research is dedicated to conducting a comprehensive exploration of the generalization abilities within deep learning. Our primary objective is to gain a deeper understanding of the underlying factors that drive generalization in deep neural networks, enabling high accuracy for simple models without the risk of overfitting as is common for complex models.** To achieve this goal, we will delve into the practical implications of various core elements, including normalization, optimization, and regularization. Through systematic practical analysis, we aim to unearth insights and formulate conclusions that contribute to an enhanced comprehension of deep network generalization. The expected significance of this work lies in its potential to:

* **Improve Model Performance**: By developing adaptive strategies for data normalization, optimization, and dropout regularization, this research can lead to improved model accuracy and reliability.
* **Enhance Generalization**: The insights gained from these investigations can contribute to models that generalize better to unseen data, making them more applicable across diverse domains.
* **Expand Knowledge**: The study of higher-order EMAs, hybrid data normalization, and multi-level dropout introduces novel concepts that expand our understanding of deep learning.
* **Real-world Applications:** The research outcomes have the potential to benefit a wide range of applications, from computer vision to natural language processing, audio analysis, and beyond.

The specific research goals that will enable us to obtain this significance are:

* ***AIM 1: Investigating Adaptive Strategy for Hybrid Data Normalization***

Our first objective revolves around exploring the potential of **hybrid adaptive data normalization techniques, incorporating knowledge-based and data-driven approaches**.

Diverging from the common fixed-grouping Group Normalization approach, our methodology harnesses the concept of channel similarity to **dynamically re-order the data**, thereby elevating both adaptability and the generalizability of the normalization process. Moreover, our framework empowers flexible adjustments in group sizes, enabling tailoring of normalization to specific data characteristics. Through this investigation, we aim to evaluate the impact, strengths, and potential limitations of this hybrid model on data normalization procedures. Additionally, we will assess the influence of our adaptive normalization on the generalization capacity of the learning model.

* ***AIM 2: Exploring High-Order Exponential Moving Average (EMA) Optimizers***

Building on the success of our recent work introducing the Triple Exponential Moving Average (TEMA) optimizer, we strive to push the boundaries further by **exploring higher-order EMA-based optimization techniques to enhance performance and stability of deep optimization**. This research endeavor entails a thorough examination of higher-order Exponential Moving Averages (EMAs) and their potential influence on the overall performance of deep learning models, with a particular emphasis on their effects on generalizability and accuracy. Our study encompasses an evaluation of the stability of these higher-order frameworks across a diverse spectrum of datasets, architectural configurations, and training scenarios. **Key research questions to be explored include whether higher-order EMAs exhibit improved generalization due to their enhanced ability to accurately identify and track gradient trends with minimal lagging, whether there exists an optimal order beyond which generalization diminishes, and whether adaptation of the optimizer order to network characteristics (such as depth) preserves generalizability.**

* ***AIM 3: Studying Multi-level Dropout Regularization for Enhanced Adaptation***

Conventional dropout techniques, alongside their adaptations, are routinely employed to deal with features co-adaptation within individual hidden units. Inspired by innovative concepts like group dropout, we propose a novel approach for dropout regularization. This technique aggregates neurons into distinct groups, guided by spatial and semantic correlations among neighboring neurons. This is a knowledge-based level. Then, we introduce a randomness-based level by randomly selecting a specific neuron from each group. That way we enjoy both worlds - we improve the dropout concept by incorporating knowledge into the procedure but at the same time by randomly selecting specific neurons from each group for mitigating overfitting, thus augmenting the overall resilience and robustness of the model. **Key research questions to be explored include whether knowledge-based regularization yields superior performance in comparison to traditional complete random dropout, and whether it enhances model generalizability or do the opposite, by introducing a novel regularization approach that is partially based on inferred knowledge, in addition to the random procedure.**

By addressing these objectives, we aim to contribute significantly to the advancement of deep learning, ultimately empowering the development of more reliable and adaptable deep learning models with broad applicability across various domains.

# Detailed Research Plan

* ***AIM 1: Investigating Adaptive Strategy for Hybrid Data Normalization***

Our research introduces a novel hybrid normalization strategy that integrates domain knowledge into data-driven learning. This pioneering approach marks a significant leap in deep learning normalization techniques. **Our model is the first to represent a unique integration of data-driven and knowledge-based techniques for data normalization.** Unlike conventional Group Normalization that relies on fixed channel-grouping, our approach leverages the concept of channel similarity to **dynamically organize the channel data**. This dynamic organization not only augments the efficiency of the normalization process but also enhances its adaptability to diverse datasets and as a result will increase generalizability. An intriguing aspect of our framework is its ability to facilitate flexible adjustments in group sizes, enabling tailored normalization that aligns precisely with the unique characteristics of the data. In addition to these core objectives, our research endeavor raises intriguing questions:

* How does the incorporation of domain knowledge into normalization impact the efficiency and adaptability of the process, and what implications does it hold for real-world applications?
* Can adaptive normalization, with its dynamic data organization and adjustable group sizes, mitigate common challenges associated with data imbalance and heterogeneous datasets, potentially revolutionizing the way we preprocess data for deep learning?
* What are the trade-offs involved in adopting normalization, and are there scenarios where traditional normalization techniques still outperform our hybrid approach?
* Does the adaptability introduced by our normalization strategy result in enhanced generalization capabilities of deep learning models, and if so, to what extent can we measure this improvement across various tasks and domains?

To do that, in our innovative Similarity-based Grouping (SGN) technique, the learning process is periodically paused every few epochs to engage in channel re-clustering, as illustrated in Figure 1.

**Figure 1:** The proposed similarity-based group normalization approach

The whole grouping procedure is done offline. It follows the steps below:

* **K-means Clustering and outliers' exclusion:** Isolation Forest algorithm is used to exclude channels that are outliers (i.e considering the mean and variance of their features), followed by K-means clustering that is calculated according to the inlier channels only. See Figure *2* for visualization of the importance of the channels clustering that is carried out in SGN.
* **Integration of the re-ordered channels in the learning process**: Having established the ***offline*** *channel clusters*, all channels are now grouped into individual clusters and are then re-integrated to the network. From that time point, the learning continues until the next re-clustering procedure.

Two different grouping strategies are explored –

* *SGN-V1* – *single mini-batch re-clustering* - clustering is applied considering all channels in a *whole mini-batch*. As a result, channels from different images can be grouped together. This allows higher flexibility, normalizing channels with similar statistics. However, this approach also introduces a constraint during inference. Given that the grouping procedure during inference should mirror the one that is applied during training, both phases require identical batch sizes. While certain applications that are based on temporal information (like voice recognition \cite{mcgehee1944experimental}, video processing \cite{liu2020deep}, and NLP \cite{nadkarni2011natural}) can manage inferences on a cluster of frames (i.e., mini-batch size greater than 1), others may face operational challenges under this restriction.
* *SGN-V2 – single image re-clustering -* re-clustering allows channels grouping within *a specific image only*. As a result, this SGN version allows testing/inference across any batch size (as it anyway normalizes each image separately), making it a preferred choice for applications with a single-image testing. On the other side, our exploration shows that this approach diminishes the potential for optimal alignment among channel group statistics. Our experimental findings indicate that SGN-V2 marginally underperforms when compared to SGN-V1.

****

1. (b) (c)

Figure 2: Each point illustrates the mean (x-axis) and the variance (y-axis) for individual channels. (a) Original GN - without clustering, channels groups are highly mixed. (b) SGN-V1. (c) SGN-V2

We perform re-clustering periodically every 10th epoch because of the following reasons:

* As training progresses, the weights gradually change their values, enhancing specific image features while diminishing others. Consequently, every few iterations, re-clustering becomes necessary due to shifts in the channel group statistics.
* Maintaining the same groupings based on initial training statistics can lead to overfitting. Regularly altering the group sequences during the restraints provides added regularization to the network, helping prevent overfitting.

Additionally, we will study two additional criteria for our analysis:

* **Unequal Groups**: traditionally, group normalization relies on fixed and equal group sizes. However, the fact that in our approach we use K-means clustering can help to generate groups of channels with varying sizes. This will help to study the accuracy and the significance of channel separation, resulting in more precise clusters.
* **STD re-calculating**: Prior to each re-clustering step, we will calculate the standard deviation (std) of channels within each group. Groups with low standard deviations will be excluded from the subsequent channels re-clustering step. In such cases, the re-clustering process will be reserved exclusively for groups exhibiting high standard deviation, surpassing a predefined threshold. This strategy ensures that resources are allocated efficiently, optimizing the re-clustering process for maximum impact.

**Preliminary results**

To evaluate the strength of our technique, we first validate it on CIFAR-100 benchmark. We employed the widely-used ResNet50 architecture, which incorporates convolutional layers and pooling layers to mitigate overfitting. The training involved 100 epochs for all models and a learning rate of 0.001 and mini-batches of 8/16 images per batch were used. Figure 3 demonstrates a comprehensive experimental analysis involving three distinct versions: SGN-V1, SGN-V2, and GN. SGN-V1 demonstrates a commendable enhancement in performance, registering a 3.15\% improvement over the original GN. Furthermore, nuanced observation reveals an additional uplift of 1.32\% in the performance of SGN-V2 compared to GN. These findings underscore the progressive refinements and superiorities of the SGN versions over the conventional GN.



Figure 3: Comparative analysis of accuracy curves for the CIFAR-100 dataset when implemented on the ResNet50 architecture. The graph illustrates the test accuracy in relation to the number of training epochs. This comparison encompasses two normalization methods: SGN V1, SGN V2, and GN. The intent is to provide a clear perspective on the performance dynamics of each method across the training process.

**Pitfalls and Alternative Approach**

We minimized the risks for pitfalls by presenting very promising preliminary results. Obviously, testing our proposed method on a larger and more diverse cohort may present some new challenges that we will have to deal with. In the event that the similarity-based method will not deliver satisfactory results, we will explore a random-based re-grouping approach, which we have already begun to explore. It is similar to our similarity-based approach that we presented above, with one main difference – here the channels will be re-grouped randomly. The motivation to do that is our assumption that generating a "mess" every *nth* epoch will enforce the learning architecture to be more robust and with stronger abilities.

* ***AIM2 – Exploring High-Order Exponential Moving Average (EMA) Optimizers***

We strongly believe that by introducing a higher-order *EMA* optimizer that is able to track gradient changes and trends much better with minimal lagging, we will be able to reduce the dependence on different algorithmic choices for optimization such as the initialization, update rules, learning rate, and stopping condition. In our recent paper [Peleg, Weiss, Hoogi], we introduced a novel TEMA-based optimizer that is based on a 3rd-order Exponential Moving Average. It is designed to accurately estimate the gradients' first and second moments to obtain less biased estimation of the true gradient at each time step, as compared to adaptive methods that use the standard *EMA*, such as Adam. This is achieved by *TEMA* 's ability to reduce lag in the estimation of the gradient moments while still performing effective denoising. The main contributions of this paper are:

* **High-order EMAs**: As far as we know, our previous paper and this extended proposal are the first to exploit the great potential of high-order Exponential Moving Averages (EMAs) for improved optimization, overcoming the limitations of first-order Exponential Moving Averages (EMA) such as inherent lag and insufficient adaptation to data trends. Our previous paper explored the third order only and here we want to investigate a series of high order EMAs
* **Active Guidance in Optimization**: In finance, Triple Exponential Moving Average (TEMA) has been used as a passive indicator, merely compared to market changes *without* directly affecting them. However, here we will explore how higher-order EMAs *actively* guide the optimization process and affect network weights, extending its applicability and usefulness in high-dimensional optimization problems.

We have compelling evidence that the higher-order Exponential Moving Average (EMA) outperforms the commonly used EMA optimizers in terms of stability, robustness, and accuracy. This finding represents a significant validation of the concepts underpinning the current grant. The initiative of this grant proposal involves a thorough investigation of higher-order EMAs and their potential impact on the overall performance of deep learning models, with a specific emphasis on their influence on accuracy and model generalization. Our study encompasses a comprehensive assessment of the stability of these higher-order frameworks across a diverse spectrum of datasets, architectural configurations, and training scenarios. We are particularly interested in exploring several intriguing research questions:

* Do higher-order EMAs exhibit improved generalization abilities owing to their enhanced capacity to accurately identify and track gradient trends?
* Is there a specific order beyond which the generalization performance starts to diminish, providing insights into the optimal order for practical use?
* Should the selection of the order be tailored to the characteristics of the network, such as its depth, to maintain consistent generalizability across various neural network architectures?

**Through this research, we aim to not only expand the EMA-based optimization but also provide valuable insights into the intricacies of selecting the appropriate higher-order EMAs for deep learning models, ultimately advancing the field's understanding of optimization techniques and their impact on model generalization.**

**Our Proposed Method**

While the simple in (1) denoises the data effectively, it also introduces lag in the estimation, which may result in inadequate gradient updates, and consequently in suboptimal performance.

For a sequence , recall that denotes the sequence obtained after applying exponential moving average on To reduce the lag, the Double Exponential Moving Average adds to the lag correcting term

 (2)

The lag correcting term in (2) can be seen as a smooth estimation of the true (but noisy) lag between the true data *x* and the naive smooth estimator , as measured by their difference . Adding the correction term in (2) to we obtain

One can go further and consider a lag correction to the first lag correction, namely

Adding the term in (4) to in (3) gives ,

When one can obtain the common-used formula for ,

In any other case, where , the equation takes on a slightly different form while retaining a similar underlying concept. Straight forward calculation shows that in this case,

where

In this research, we will introduce and explore the generalization of *EMA* to general order *EMA*, call it as follows. denote the recursive application of *EMA* on for times. We will also define

 (9)

and for *k=1,2,3* …

 (10)

Let and for *k=1,2,3*…

 (11)

 can be seen as -order lag correction operator. The calculation below shows that

Higher-order *KEMAs* sacrifice more smoothness in exchange to more aggressive lag reduction. Thus, in practice the order should be selected to appropriately balance between reducing the lag on the one hand and denoising the noisy gradients on the other hand. The appropriate order highly depends on the problem at hand. Note that in general, additional *EMAs* at each level introduce additional hyper-parameters to be tuned. Evidently, a larger *k* entails larger coefficients in the expansion for , making the high-order less stable to hyper-parameters tuning. This affects too the choice of order one should use. From the linearity of the *EMA* operator we may w.l.g. expand and as the sums

By definition

 *(12)*

Defining for convenience and plugging in the last sum into (8) gives

Matching the coefficients of in (13) to that in (10) gives the recursive equation,

Recalling that by definition and , the solution to (14) is as in (10), since

To show the expansion of in (9), we have

Matching the coefficients of to that in (11), we obtain the recursive equation,

Since and by definition , we obtain that the solution is as in (9) since

According to the developed equations, we will study the performance of high-order optimizers, while carefully explore their stability that is affected by . In a first sight, we expect that higher-order might be a bit unstable because of scale which is bigger than the one obtained for *TEMA*

***Preliminary results (these results are the results from our TEMA-based paper, we didn’t start to work on the higher-order KEMAs, is there a way to include them as well?)***

By now, we have conducted an extensive validation of our preliminary TEMA-based optimizer, subjecting it to rigorous testing across a wide range of domains. Our evaluation spanned five diverse datasets (CIFAR-10, CIFAR-100, PASCAL-VOC, MS-COCO, and Cityscapes), encompassing 14 distinct architectural configurations, and addressing various computer vision tasks, including object detection, image classification, and semantic understanding. TEMA-BASED OPTIMIZER underwent thorough comparison with six different optimization methods, namely SGD with momentum, Adam, AdamW, AdaHessian, AdaBound, and AdaGrad. To ensure the reliability of our findings, we averaged the results over two different weight initializations.

*CIFAR-10 and CIFAR-100*

Our TEMA-based optimizer exhibited significant superiority over both Adam and SGD in 83.3% of the tested architectural setups on the CIFAR-10 and CIFAR-100 benchmarks. Furthermore, it outperformed other optimizers such as AdaBound, AdamW, AdaGrad, and AdaHessian, consistently achieving the highest accuracy. Across all architectures analyzed for CIFAR-10/100, our TEMA-based enhanced the average classification accuracy by 1.16% (AdamW), 1.57% (AdaBound), 3.52% (AdaHessian), and a remarkable 16.33% (AdaGrad). Additionally, our optimizer showcased smoother test-accuracy curves, reducing the impact of random gradient estimation fluctuations compared to standard Exponential Moving Average (EMA), thanks to TEMA's less biased estimation of the true gradient.

*Pascal-VOC*

Our study demonstrated that our TEMA-based optimizer surpassed all the compared optimizers (SGD, Adam, and AdamW) in optimizing the YOLOv5-s and YOLOv5-m architectures, as evidenced by higher values for mean Average Precision (mAP) at an Intersection over Union (IoU) threshold of 0.5. We also improved mAP@0.5 by 6.56% in YOLOv5-s and by 7.76% in YOLOv5-m, highlighting its exceptional performance in optimizing models for object detection.

*MS-COCO*

Our optimizer significantly outperformed SGD, Adam, and AdamW in optimizing the YOLOv5-n architecture across all key statistical parameters (mAP@0.5, Precision, Recall, and F1-score). It achieved an average improvement of 16.9% for mAP@0.5, 16.1% for Precision, 15.7% for Recall, and 15.16% for F1-score. These results underscore the remarkable effectiveness of our optimizer in enhancing the performance of object detection models.

*Cityscapes benchmark*

Results show that our optimizer improves the mean IoU results of SGD, Adam and AdamW by 5.3%, 1.4% and 1.9% respectively.

In conclusion, our extensive experimentation and comparisons across a wide array of datasets, architectures, and computer vision tasks unequivocally establish the consistent superiority of our optimizer in delivering state-of-the-art results. This positions it as a valuable tool for optimizing neural network models across diverse domains. **Furthermore, our comprehensive experimental setup underscores our optimizer's robustness and showcases its ability to provide less noisy gradient predictions. Importantly, it demonstrates that the accuracy achieved with our simple network is on par with, if not superior to, the accuracy attained by other optimizers using more complex models from the same family (e.g., ResNet or DenseNet).**

* ***AIM3 - Studying Multi-level Dropout Regularization for Enhanced Adaptation***

Expanding upon the notion of adaptive dropout, our research introduces an innovative adaptive model that leverages both knowledge and randomness to accommodate varying data scales. **This research aims to shed light on the nuanced interplay between regularization techniques and model generalization, delving into the inherent strengths of randomness in contrast to incorporating knowledge or to the advantage of their combination.**

Our approach consists of three main steps:

* We employ a hybrid framework that combines Convolutional Neural Networks (CNNs) with Neural Additive Model (NAM) [REF] to enhance our model's performance. Specifically, our current architecture utilizes ResNet-34 as the foundational CNN structure. Within the middle layer of the ResNet, we seamlessly integrate the NAM model. This integration of NAM serves a dual purpose:

1. **Fine-Grained Neuron Analysis**: NAM provides a valuable mechanism for discerning the individual contributions of neurons within the network. This level of granularity enables us to gain deeper insights into how specific neurons influence the model's predictions, enhancing our understanding of the network's inner workings.

2. **Parameter Efficiency**: An added advantage of incorporating NAMs lies in their ability to substantially reduce the number of network parameters. This reduction in model complexity results in more efficient computations, facilitating faster training and inference times.

* We will employ a similarity-based clustering according to the significance of individual neurons for the given task and their prominence within a particular feature map.
* We will then apply a random selection of a predetermined percentage of neurons from each cluster to undergo dropout in each training epoch.

**Our first hybrid random-knowledge dropout approach ensures that the dropout procedure will not be fully random, but it will have a rationale behind it.**

We will explore the following aspects:

* Knowledge-Enhanced Generalization: To what extent does the knowledge-driven grouping and dropout strategy enrich the learning model's performance by preserving crucial feature representations? Does this approach lead to more consistent and superior generalization compared to conventional random dropout, and if so, in what scenarios or network architectures is this effect most pronounced?
* Stability and Overfitting Mitigation: Does the carefully introduced randomness within the adaptive group dropout framework effectively mitigate overfitting to the training set? How does this approach strike a balance between model stability and the risk of increasing overfitting? Can it enhance the model's resistance to noise and perturbations in real-world data?
* Optimal Grouping Strategies for Robust Generalization: What are the key factors influencing the success of different neuron grouping strategies? Are there specific spatial or semantic relationships among neurons that are particularly conducive to robust generalization? How do these strategies adapt to varying dataset characteristics and network complexities?
* Transferability Across Domains and Tasks: To what degree can the benefits of adaptive group dropout be transferred across different domains, datasets, and tasks? Does this technique offer a transferable improvement in the generalization ability of deep learning models, and are there limitations to its applicability in specific contexts?

**By addressing these refined research questions, our study aims not only to advance the understanding of adaptive group dropout but also to provide actionable insights into its role as a potent regularization tool for enhancing model generalization.** We aspire to unravel the intricate interplay between knowledge-driven dropout strategies, model stability, overfitting mitigation, and the broader context of deep learning, contributing to the ongoing evolution of techniques that empower deep neural networks to generalize effectively across diverse datasets and real-world scenarios.

**Preliminary results**

Preliminary results show a significant improvement of 1.5% in the classification accuracy for CIFAR-100 when applying a ResNet34 network. Results also show a significant 70% decrease in the number of parameters and computational load of the learning architecture.

**Should be extended**

**Pitfalls and alternative approaches**

**Still have to be filled**