

Article

A Comprehensive Study of MCS-TCL: Multi-Functional Sampling for Trustworthy Compressive Learning

Fuma Kimishima , Jian Yang  and Jinjia Zhou * 

Graduate School of Science and Engineering, Hosei University, Koganei Campus, Tokyo 184-8584, Japan; fuma.kimishima.3c@stu.hosei.ac.jp (F.K.); jian.yang.4f@stu.hosei.ac.jp (J.Y.)

* Correspondence: zhou@hosei.ac.jp

Abstract

Compressive Learning (CL) is an emerging paradigm that allows machine learning models to perform inference directly from compressed measurements, significantly reducing sensing and computational costs. While existing CL approaches have achieved competitive accuracy compared to traditional image-domain methods, they typically rely on reconstruction to address information loss and often neglect uncertainty arising from ambiguous or insufficient data. In this work, we propose MCS-TCL, a novel and trustworthy CL framework based on Multi-functional Compressive Sensing Sampling. Our approach unifies sampling, compression, and feature extraction into a single operation by leveraging the compatibility between compressive sensing and convolutional feature learning. This joint design enables efficient signal acquisition while preserving discriminative information, leading to feature representations that remain robust across varying sampling ratios. To enhance the model's reliability, we incorporate evidential deep learning (EDL) during training. EDL estimates the distribution of evidence over output classes, enabling the model to quantify predictive uncertainty and assign higher confidence to well-supported predictions. Extensive experiments on image classification tasks show that MCS-TCL outperforms existing CL methods, achieving state-of-the-art accuracy at a low sampling rate of 6%. Additionally, our framework reduces model size by 85.76% while providing meaningful uncertainty estimates, demonstrating its effectiveness in resource-constrained learning scenarios.

Keywords: compressive learning; compressive sensing; image classification; evidential deep learning



Academic Editor: Kohei Arai

Received: 22 July 2025

Revised: 31 August 2025

Accepted: 5 September 2025

Published: 7 September 2025

Citation: Kimishima, F.; Yang, J.; Zhou, J. A Comprehensive Study of MCS-TCL: Multi-Functional Sampling for Trustworthy Compressive Learning. *Information* **2025**, *16*, 777. <https://doi.org/10.3390/info16090777>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the advancement of smart societies in recent years, image sensing technology has undergone significant evolution. The signal processing domain has changed from analogue to digital, enabling more flexible and cost-efficient processing. Within this approach, signals are typically sampled at or above the Nyquist rate to ensure high-fidelity reconstruction. However, when considering real-world applications, the Nyquist rate often becomes excessively high, posing challenges for practical implementation due to hardware constraints.

Compressive sensing (CS) [1] enables more efficient signal processing and has attracted significant attention from both research and industry. In the sampling process, the input signal is acquired by a linear random projection and then lower-dimensional vector representations, called measurements, are observed. CS theory guarantees that

the original signal can be reconstructed from these measurements if the input signal is sparse in some regard, and that this can be achieved with far fewer measurements than the Nyquist sampling rate requires [2]. The merit of applying these features to practical problems includes freeing up hardware computer memory and reducing the burden of communication while ensuring perfect reconstruction. To fully enjoy these benefits, CS has mainly been applied in computer vision fields such as MRI [3], single-pixel imaging [4,5], snapshot compressive imaging [6], and so on. The value provided by these imaging systems is determined based on visual quality. Therefore, amongst CS scholars, attention has recently been focused on signal reconstruction methods, and certain insights have been provided over the past decade regarding mathematical theories on reconstructing images from measurements [7–10], as well as the integration of deep learning [11–15].

CS is assumed to be applied not only to provide attractive visuals but also to analyse images to respond to specific needs, such as image classification. In such cases, there is no need to prioritize high-quality image reconstruction because the focus of interest is the inference result. In fact, reconstruction may even be undesirable in specific applications [16] due to the possibility of leaking personal information. The concept born from this demand is compressive learning (CL) [17], which skips the reconstruction phase and executes the inherent task directly from measurements. To adapt to this new framework, various attempts have been proposed, including modifications to sampling components [18,19], extraction of discriminative features from measurements [20], and system design to jointly optimize the sampling matrix and task-specific networks [21,22]. In particular, MCL [23], which considers the multidimensional nature of the input signal, and TransCL [24], which incorporates ViT [25] as a task-specific network, have been the most successful methods in recent years, suggesting the potential for new CS applications.

As CL is primarily designed for vision-related tasks, many existing frameworks incorporate a module that transforms compressed measurements into image-like representations to better suit downstream processing. However, this approach presents two major challenges. First, these methods often increase the dimensionality of the compressed measurements in an attempt to compensate for the information loss incurred during sampling. Since the original signal is not fully preserved in the measurements, this process effectively resembles implicit reconstruction. Feeding such expanded data into task-specific networks increases model complexity and memory requirements, which significantly limits the practicality of CL in resource-constrained environments. Second, these frameworks typically overlook the uncertainty inherent in inference from incomplete information. The reconstructed images derived from compressed measurements are often noisy and lack sufficient detail about the original inputs [23]. Moreover, in real-world scenarios, additional degradation may occur during transmission or processing. Consequently, the resulting predictions can be unreliable, posing significant concern for applications where trustworthy inference is essential.

To address the aforementioned challenges, we propose a novel framework called Multi-functional Compressive Sensing Sampling-based Trustworthy Compressive Learning (MCS-TCL). This approach integrates image sampling, compression, and feature extraction into a unified CS-sampling process and directly produces inference results while accounting for prediction uncertainty. Unlike existing CL methods that expand the dimensionality of measurements, MCS-TCL rearranges measurement vectors into structured image representations without dimensional expansion, resulting in a suitable input format for the classifier and enabling efficient inference with reduced computational cost. Furthermore, this process is inherently compatible with convolutional operations, allowing the transformed representations to function as effective feature maps. As a result, meaningful image features can be expressed in a low-dimensional form that remains robust across different sampling

rates (SRs), thereby enhancing the overall reliability of CL. Furthermore, to ensure the trustworthiness of the inference, MCS-TCL incorporates an evidential deep learning (EDL) framework [26] to quantify prediction uncertainty. We also introduce a novel loss function that encourages the model to accumulate more evidence for the correct class, enabling it to make more confident and interpretable predictions. The main contributions of this work are summarised as follows:

- We propose a novel framework, Multi-functional Compressive Sensing Sampling-based Trustworthy Compressive Learning (MCS-TCL), which unifies image sampling, compression, and feature extraction within a single process while explicitly modelling uncertainty in the final predictions.
- To enable direct task execution from compressed measurements without explicit reconstruction, we introduce a method that transforms measurement vectors into structured image-like representations, allowing for seamless integration with convolutional operations.
- We incorporate evidential deep learning (EDL) into the task-specific network and propose a tailored loss function that effectively quantifies prediction uncertainty and encourages the model to assign stronger evidence to the correct class.
- Experimental results on image classification tasks demonstrate that MCS-TCL achieves state-of-the-art performance at a low sampling rate while reducing model size by 86.57%, highlighting both its effectiveness and efficiency.

A preliminary version of this work was previously presented [27], where we introduced a brief overview of the proposed framework, MCS-TCL. This paper significantly expands on this by providing a complete methodological description, in-depth analysis, and extensive empirical evaluations. As such, it presents a self-contained and substantially enriched contribution.

2. Related Works

2.1. Compressive Sensing

Compressive sensing (CS) [1] is a new signal processing paradigm that can recover original signals from a significantly lower-dimensional linear projection than conventionally required. Unlike traditional signal acquisition methods governed by the Nyquist sampling theorem, CS allows for exact signal reconstruction with high probability when the signal exhibits sparsity in a suitable transform domain.

Mathematically, given the original signal $x \in \mathbb{R}^N$ and sampling matrix $\Phi \in \mathbb{R}^{M \times N}$ under $M \ll N$, the measurement $y \in \mathbb{R}^M$ can be expressed by $y = \Phi x$. The sampling ratio (SR), which indicates how many signals are sampled, can be defined as $\frac{M}{N}$. Recovering the original signal x from the low-dimensional y is an ill-posed problem. To perform a solid reconstruction, CS-based methods usually require the following optimization problem to be solved:

$$\min_x \frac{1}{2} \|\Phi x - y\|_2^2 + \lambda \phi(x) \quad (1)$$

where $\phi(x)$ represents a transformation of x under a suitable transform operator, $\phi(\cdot)$, and this term is regarded as a prior with regularised parameter λ . To solve this problem, various approaches have been proposed that involve mathematical methods such as Basis Pursuit (BP) [28] and Iterative Shrinkage-Thresholding Algorithm (ISTA) [29].

With the advancement of neural networks, recent research has actively explored solving this problem using deep learning-based approaches; these are generally regarded as deep non-unfolding networks [30–33], whereas deep unfolding networks [6,12–15,34], which integrate neural networks with mathematical optimization algorithms, have also

gained significant attention. However, when considering the practical application of CS in real-world scenarios, achieving high-quality reconstruction may not always be the primary objective. In tasks such as image recognition and classification, inference performance is often more important than the fidelity of the reconstructed image.

2.2. Compressive Learning

Calderbank et al. [17] and Davenport et al. [35] proposed the compressive learning (CL) framework, which performs inference tasks directly from measurements without signal reconstruction. The authors of [36] revealed that, with a sufficiently large random sampling matrix, the structure of a signal can be captured. CL has been applied to image classification [35], activity recognition [37,38], and face recognition [20]. Theoretical CL research has also been developed in addition to studies on its applications. Refs. [17,39] showed that the performance of an SVM trained on measurements was comparable to that of the best linear threshold classifier trained on original signals under the constraints imposed on the sampling matrix. Reference [19] provides a measurement design for the Gaussian mixture model, and [18] extends it by learning the sampling matrix. The authors of [40] discuss the key limitation of CL—lower bounds for the probability of error—in classifying sparse signals on a standard canonical basis. The authors of [41] show that the misclassification probability decays exponentially with the number of measurements. Advancements in optimization techniques have led to the development of end-to-end CL [21,22,42] that simultaneously optimises the sampling matrix and task-specific networks. The following works [43,44] show that optimising the sensing component and the classifier simultaneously time provides superior performance.

In recent years, MCL [23] and TransCL [24] have gained significant attention for their outstanding performance in tasks such as image classification and segmentation. MCL [23] takes into account the multidimensional nature of input signals and treats the measurements as tensors rather than vectors. This approach successfully preserves the expressive power of the input signal, leading to superior performance in image classification tasks compared to existing methods while also achieving computational memory efficiency. The most recent work, involving what is known as TransCL [24], employed a ViT as a classifier, enhancing its representational capacity to achieve high-accuracy outputs even with limited information. As a result, TransCL demonstrated competitive classification performance compared to the image domain, suggesting that the task-specific network plays a crucial role in the overall effectiveness of the CL framework.

However, existing methods incorporate a mechanism to expand the dimensions of the measurements to achieve sufficient accuracy for the task, leading to larger model sizes. Moreover, since inference is performed on signals with some degree of missing information, task-specific networks can produce uncertain prediction results. Despite such circumstances, there are no existing works that consider this phenomenon.

2.3. Evidential Deep Learning

In many classification problems, the softmax function is used to calculate class probabilities. However, it is prone to over-confidence on incorrect predictions [45], and its output is the point estimate of the probability distribution [46], making it unable to estimate the uncertainty of predictions. To solve the above issues, evidential deep learning (EDL) [26,47] is proposed based on Dempster–Shafer theory (DST) [48] and subjective logic theory [49]. DST is a generalization of Bayesian theory to subjective probability. Because uncertainty estimation is relatively easy, it is used in a wide range of fields such as audio-visual event perception [50], several classification problems [51–53], social event detection [54,55], and so on. DST assigns belief weights to the set of possible class labels for a prediction and

infers the true class from these sets. By assigning all belief weights to all classes, the state of no possibility, i.e., “I don’t know (= uncertainty)”, can be expressed [26]. Mathematically, assuming there are K classes, the sum of $K + 1$ values is one, as follows:

$$u + \sum_{k=1}^K b_k = 1 \quad (2)$$

where u and b_k represent the uncertainty weight and the belief weight for k -class, respectively. To assign the belief weights, subjective logic theory uses the Dirichlet distribution, which is defined as follows:

$$Dir(\mathbf{p}|\boldsymbol{\alpha}) = \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^K p_k^{\alpha_k-1} \quad (3)$$

where $p_k \in [0, 1]$ represents the probability of belonging to class k , $\boldsymbol{\alpha}$ is a parameter for the Dirichlet distribution and $B(\cdot)$ is the beta function. The belief and uncertainty weights can be expressed as follows:

$$b_k = \frac{\alpha_k - 1}{S}, u = \frac{K}{S} \quad (4)$$

where α_k is the Dirichlet parameter for the k -th class, and $S = \sum_{k=1}^K \alpha_k$ and b_k can be considered as the probability for the k -th class. This design prevents the class probability from containing a degree of uncertainty and enables explicit measurement.

DST defines the degree of support for a sample belonging to a specific class as evidence. In the neural network, the evidence for each class $\mathbf{e} = [e_1, e_2, \dots, e_K]$ is the output obtained by a non-negative activation layer, such as an exponential function, not a softmax layer. By setting $\alpha_k = e_k + 1$, the parameters of the Dirichlet distribution depend on the evidence. As a result, the belief and uncertainty weight obtained from Equation (4) can be interpreted as evidence-based. One of the main advantage of EDL is its ability to predict both class probabilities and associated uncertainty, allowing models to quantify uncertainty reliably even under distribution shifts or adversarial perturbations. In this paper, we focus on solving the uncertainty problem in CL by leveraging evidential deep learning techniques.

3. Proposed Methods

In this section, we present the proposed MCS-TCL framework, as illustrated in Figure 1. The framework simultaneously performs feature extraction and compression during the image sampling process and generates inference results with explicit consideration of predictive uncertainty. We begin by introducing a method to transform the measurement vector into multiple structured image representations. We then show that this transformation is compatible with convolutional operations and that the resulting structures can be interpreted as feature maps. Next, we describe the advantages of the proposed training scheme. Finally, we revisit a commonly used loss function in evidential deep learning (EDL), highlight its limitations, and introduce a new loss function that encourages stronger evidence accumulation for the target class.

3.1. CS Sampling with Measurement Transformation

3.1.1. Algorithm Conversion from Measurements to Feature Maps

Block-based CS (BCS) [56,57], which divides the input image into several small square blocks and samples from each block, is a mainstream sampling technique due to its small sampling area, allowing for easy and fast signal acquisition. We use it to obtain measurement vectors and transform them into several images, as shown in Figure 2. First, using $N \times N$ blocks, input image $\mathbf{X} \in \mathbb{R}^{H \times W}$ is divided into $B (= \frac{H}{N} \times \frac{W}{N})$ blocks. Then, each block $\mathbf{x}_b$ ($1 \leq b \leq B$) is sampled according to the CS sampling rule. Specifically, the mea-

surement vector $y_b \in \mathbb{R}^{M \times 1}$ is obtained by multiplying the sampling matrix $\Phi_b \in \mathbb{R}^{M \times N^2}$ ($1 \leq M \leq N^2$) and the block $x_b \in \mathbb{R}^{N^2 \times 1}$. By performing this operation for all blocks, B measurement vectors are obtained.

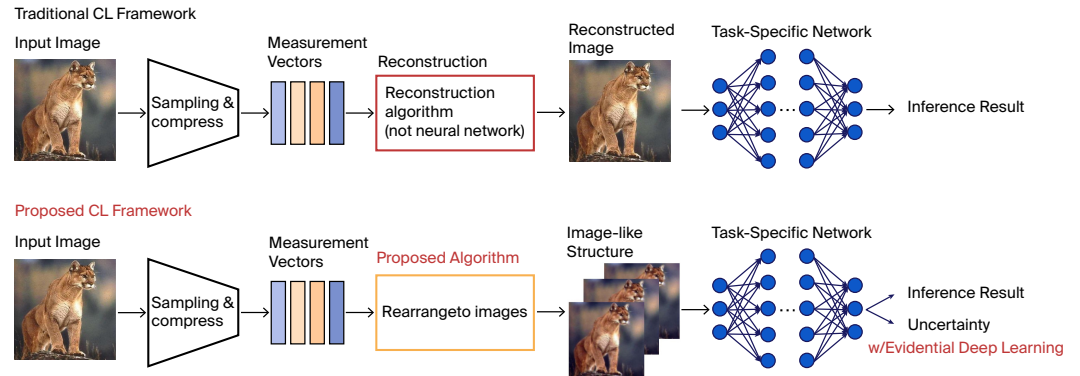


Figure 1. Diagrams of the traditional CL framework (**above**) and the proposed CL framework (**below**). Traditionally, reconstruction is performed to compensate for information for inference. In contrast, we rearrange the measurement to achieve an image-like structure without extended dimensions. This prevents an increase in the model size of the task-specific network. Furthermore, to address concerns regarding ambiguous predictions, we employ evidential deep learning (EDL) to quantify uncertainty.

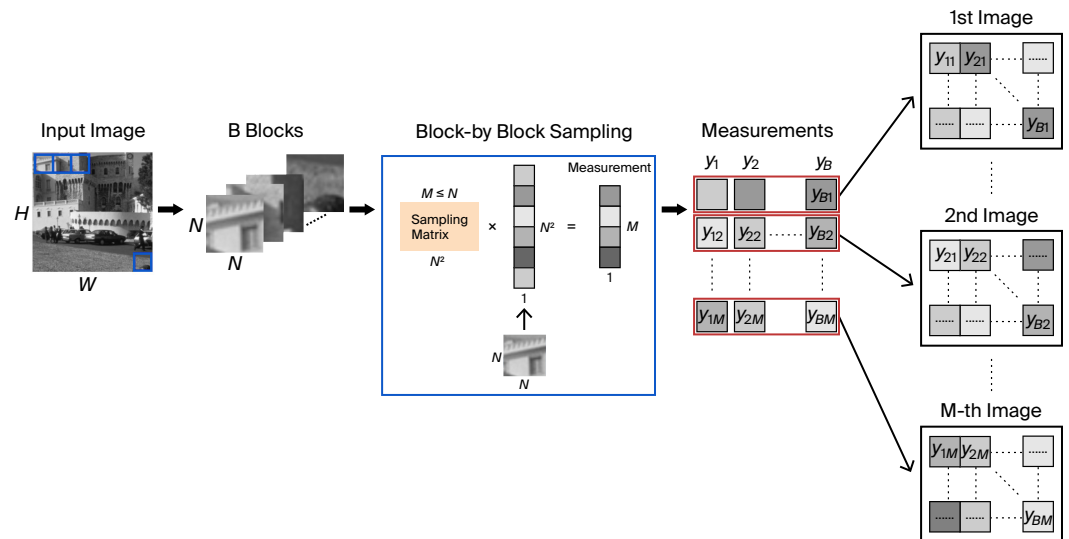


Figure 2. A series of processing flows from sampling to converting measurement vectors into feature maps. The blue boxes represent block-wise CS sampling. The values of the measurement vector enclosed in the red boxes correspond to the elements of a single image structure.

We transform all measurement vectors into several image structures. To achieve this, the i -th ($1 \leq i \leq M$) elements of B measurement vectors are grouped to form a two-dimensional structure of $\sqrt{B} \times \sqrt{B}$, as circled by the red line in Figure 2. By repeating this process for M rows, M image structures are obtained. This series of operations constitutes an algorithm that converts measurement vectors into several image structures. In the next subsection, we describe the validity of this algorithm and the details of the obtained image structures.

3.1.2. Validity of Algorithm

We describe the structural characteristics of the image representations generated by the aforementioned algorithm. As illustrated in Figure 3, the proposed method reshapes each top-left image block into a one-dimensional vector, which is then multiplied by the

first row of the sampling matrix. The resulting scalar value is assigned to the top-left position of a $\sqrt{B} \times \sqrt{B}$ image structure. This procedure is repeated for all M rows of the sampling matrix, populating the top-left positions of M image structures. The image block is then shifted to the right, and the same operations are applied to fill the next column positions of each structure. This process continues by scanning the image block across the entire spatial domain.

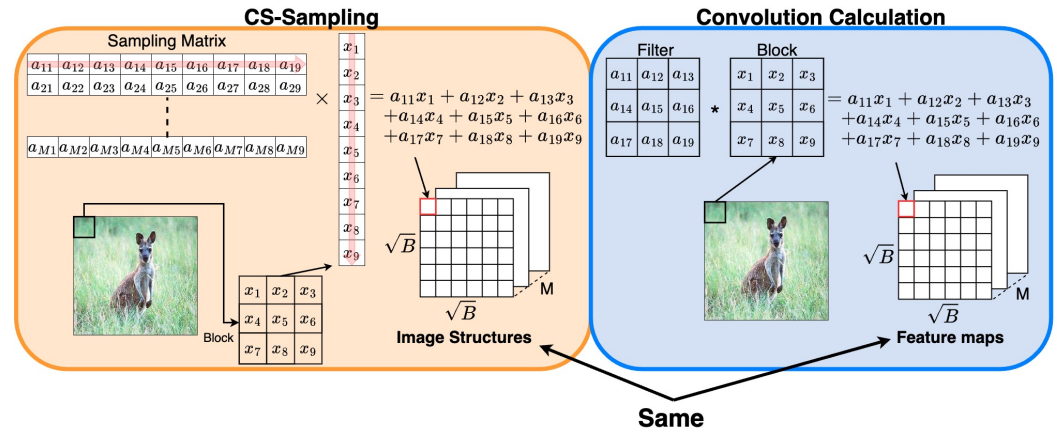


Figure 3. Illustration of the CS-sampling algorithm (left) and convolution calculation (right).

A key insight is that this computation closely resembles the operation of a convolution [58]. In standard convolution, each filter computes the sum of element-wise products between its weights and a local image patch, generating feature map values as the filter slides over the image. Similarly, in the proposed method, the image blocks serve as sliding windows, and the sampling matrix rows play a role analogous to convolutional filters. As the blocks shift across the image, the resulting values populate spatially organised structures, effectively forming feature maps.

While CS is traditionally known for simultaneously sampling and compressing signals by reducing dimensionality, our framework extends this notion. By interpreting CS sampling as a convolution-like operation, our method also performs feature extraction during sampling, adhering to the CS framework. Moreover, the extracted features are organised in a 2D image format, rather than as a flat vector, making them inherently compatible with downstream computer vision tasks. In the following subsection, we provide a detailed analysis of the resulting feature maps and explain how they contribute to efficient and effective model training.

3.1.3. Characteristics of Generated Feature Maps

In this section, we explain the properties of feature maps and the benefits when training the task-specific network. First, we introduce the relationship between SR and the size of feature map. As mentioned above, the size is $\sqrt{B} \times \sqrt{B} = \sqrt{\frac{H}{N} \times \frac{W}{N}} \times \sqrt{\frac{H}{N} \times \frac{W}{N}}$. Under $H = W$, it can be transformed as follows:

$$\begin{aligned} \sqrt{\frac{H}{N} \times \frac{W}{N}} \times \sqrt{\frac{H}{N} \times \frac{W}{N}} &= \sqrt{\frac{H^2}{N^2}} \times \sqrt{\frac{H^2}{N^2}} \\ &= \frac{H}{N} \times \frac{H}{N} \end{aligned} \quad (5)$$

From this formulation, it can be observed that the spatial size of the resulting feature maps is determined based on the block size N . As N increases, the resolution of the output feature maps decreases accordingly. Although smaller feature maps may lack fine-grained

details, the number of generated feature maps increases with larger N , since the number of sampling rows M can range from 1 to N^2 . Consequently, a higher number of feature maps can compensate for the loss of spatial detail, potentially enhancing performance through redundancy and diversity in features.

Next, we describe the training scheme differences when the CS-sampled data is represented as either a vector or an image. In the vector-based representation, the measurement vector of length M is treated as an indivisible unit—each element lacks spatial or semantic independence. Therefore, existing CL methods typically require training separate models for each SR, as the model must adapt to different input dimensions. In contrast, our image-based representation treats each feature map as a semantically meaningful entity. Since the proposed method generates N^2 image-like feature maps from the input, it is possible to select any subset of M maps during inference to simulate different SRs. Thus, the model can be trained once using the full set of N^2 feature maps (i.e., SR = 1), and once testing occurs, any arbitrary SR can be simulated by selecting a corresponding subset. This strategy greatly reduces the training cost and enhances the model's flexibility across multiple sampling rates.

3.2. Estimating the Uncertainty

The algorithm described in the previous subsection generates multiple feature maps, which are then inputted to the inference network. It should be noted that the generated feature maps do not retain all the features of the input image. In addition, taking into account the transfer of these images, noise may be added and further degrade the information. Since inference is performed directly from the degraded information, the model may output uncertain predictions. Therefore, the uncertainty should be quantified to elucidate the clarity of model. In our task-specific network, we quantify the uncertainty of predictions according to EDL.

The loss function widely used in the EDL framework is the Bayes risk of cross-entropy loss $\mathcal{L}_{ce}$ [26], which can be expressed as follows:

$$\begin{aligned}\mathcal{L}_{ce} &= \int \left[\sum_{k=1}^K -y_k \log p_k \right] \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^K p_k^{\alpha_k - 1} dp \\ &= \sum_{k=1}^K y_k \left[\psi(S) - \psi(\alpha_k) \right]\end{aligned}\quad (6)$$

where y_k is the one-hot vector of the class label and $\psi(\cdot)$ is the digamma function. This loss function focuses on assigning more evidence to the correct class. However, there have been no efforts to reduce the evidence assigned to incorrect classes. As a result, surplus evidence is generated, leading to increased belief weight and unjustifiably reduced uncertainty. We seek to determine the loss function that controls the evidence dispersed to incorrect classes to properly quantify uncertainty and effectively utilize surplus evidence.

To solve this weakness, we introduce the distribution $\tilde{e}$, which adds evidence assigned to incorrect classes to the target class, as shown in Figure 4. By computing the Kullback–Leibler (KL) divergence between the original evidence distribution e and the transformed distribution $\tilde{e}$, we penalize evidence assigned to incorrect classes. This encourages the model to concentrate more evidence on the correct class while reducing spurious

confidence in incorrect predictions. Mathematically, this calculation can be expressed as follows:

$$\begin{aligned}\mathcal{L}_{kl} &= KL(Dir(\mathbf{p}|\boldsymbol{\alpha})||Dir(\mathbf{p}|\tilde{\boldsymbol{\alpha}})) \\ &= \log \frac{\prod_{k=1}^K \Gamma(\alpha_k) \Gamma(\tilde{S})}{\Gamma(S) \prod_{k=1}^K \Gamma(\tilde{\alpha}_k)} + \sum_{k=1}^K (\tilde{\alpha}_k - \alpha_k) (\psi(\tilde{\alpha}_k) - \psi(\tilde{S}))\end{aligned}\quad (7)$$

where $\tilde{\alpha}_k = \tilde{e} + 1$, $\tilde{S} = \sum_{k=1}^K \tilde{\alpha}_k$ and $\Gamma(\cdot)$ is the gamma function.

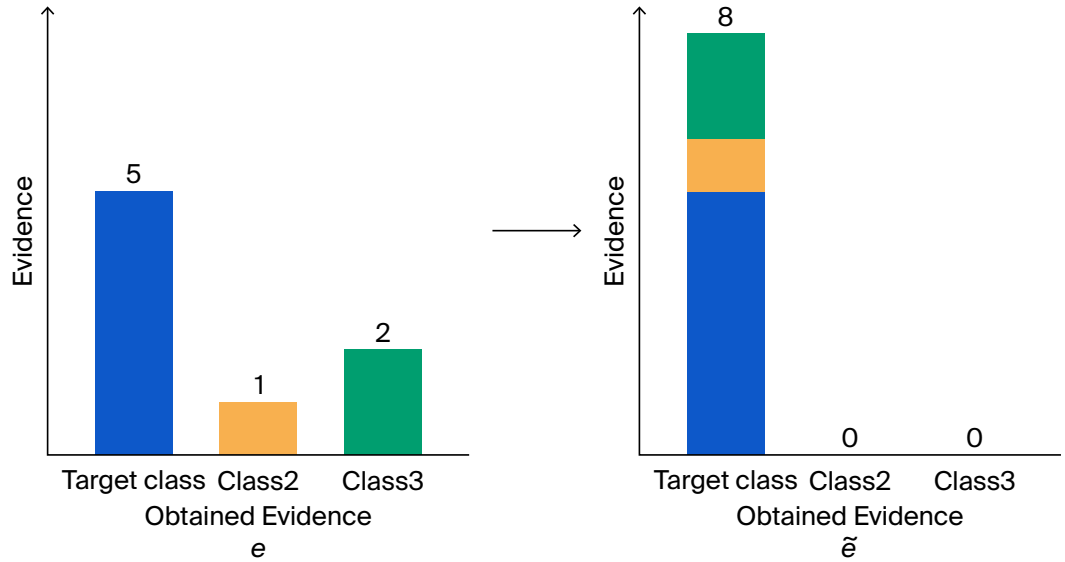


Figure 4. The evidence distribution e obtained in three classifications (left) and the evidence distribution $\tilde{e}$ obtained by integrating evidence assigned to the incorrect classes into the target class (right).

To impose an orthogonal constraint on the sampling matrix Φ_b , $\mathcal{L}_{orth}$ is defined as $\mathcal{L}_{orth} = \frac{1}{M^2} \|\Phi\Phi^T - I\|_F^2$ where I represents the identity matrix. The entire loss function $\mathcal{L}_{all}$ can be expressed as follows:

$$\mathcal{L}_{all} = \mathcal{L}_{ce} + \lambda_t \mathcal{L}_{kl} + \gamma \mathcal{L}_{orth} \quad (8)$$

where $\lambda_t = \min\{0.01, \frac{t}{1000}\}$ is the annealing coefficient, t is the current epoch and γ is the regularization parameter, which is set as 0.001 in our experiment. In our framework, the sampling process and the task-specific network are jointly optimised by simultaneously computing the constraints of the sampling matrix and the loss of the inference results.

4. Experimental Results

We demonstrate the superiority of our proposed framework, MCS-TCL, in image classification tasks. To evaluate its performance, we compare it with state-of-the-art CL methods including MCL [23] and TransCL [24]. Since our approach can output a score representing the degree of uncertainty in predictions, we are able to provide insights on how ambiguous the classification model perceives a given sample to be. Furthermore, we use datasets with larger image sizes and relatively fewer samples, aligning with real-world social applications.

4.1. Datasets

As Chong et al. [24] pointed out, existing CL methods have been validated on datasets containing small images such as MNIST and CIFAR [59,60]. However, real-world applications often require high-resolution image processing and, thus, complex computations

involving large sampling matrices, leading to significant hardware burdens and potential concerns as to whether the processing speed can meet user expectations. Thus, the applicability of existing CL methods is limited. In our experiments, we demonstrate that MCS-TCL can process high-resolution images by employing three datasets: Caltech101 [61], the UC Merced Land Use Dataset [62], and RESISC45 [63]. Note that all datasets were used under standard sensing conditions, where the conversion from measurements to feature maps is well defined.

1. Caltech101 contains 8677 images from 101 classes. The image size is roughly 300×200 pixels. The number of images included in each category varies from 40 to 800.
2. UC Merced Land Use Dataset has 21 classes of objects, with each class consisting of 100 images. All image sizes are 256×256 pixels.
3. RESISC45 consists of 31,500 images. For each of the 45 classes, there are 700 images with a size of 256×256 pixels.

Since TransCL resizes images to 384×384 before inputting them into the task-specific network, we adopt the same resizing strategy for both MCL and our proposed model to ensure consistency in evaluation. Although it is technically feasible to use 32×32 images and upscale them to 384, this kind of approach introduces noise due to the upsampling process, which can adversely affect both model training and inference. To mitigate these potential issues, we utilize datasets containing higher-resolution images, ensuring a more robust and reliable evaluation of our method. All experiments are conducted under standard sensing conditions, i.e., assuming regular spatial structures for the conversion from measurements to feature maps, while handling non-uniform or adaptive sensing is left as future work.

4.2. Metrics

We use top-1 accuracy (ACC) and area under curve (AUC) as our chosen evaluation metrics [64]. In our work, the last layer in the task-specified network is set as the EDL layer, which is capable of outputting uncertainty. To evaluate model size, we utilize metrics considering the number of parameters, model size, and multi-adds. Moreover, we show the number of measurements (Meas) input into the task-specified neural networks.

4.3. Implementation Details

In this experiment, the classifier for all frameworks is set as a vision transformer (ViT) pre-trained on ImageNet-1K. The configuration of the ViT is the same as that of TransCL [24], consisting of 12 transformer layers and 12 heads, and the dimension of the embedding is 768. We use the SGD optimizer with a momentum of 0.9 and weight decay of 0.0005. The initial learning rate is 0.003, utilising the cosine annealing strategy [65]. We set the batch size and the number of epochs as 256 and 100, respectively. As with TransCL [24], data augmentation is employed via random horizontal flipping during training. We implement MCS-TCL with PyTorch with the version 2.7.1 [66] on an RTX3090 GPU. In our experiments, we set small SRs equal to 6.25%, 12.5% and 25%.

4.4. Ablation Study

4.4.1. Effectiveness of Block Size

The visual changes in the generated feature maps when changing the block size, B , are illustrated in Figure 5. We resized all images to 384×384 for easier comparison. It can be observed that as the block size increases, the image size decreases, leading to a loss of information on object characteristics, such as edges and textures. When inputting these feature maps into a task-specific network, it cannot retain the features of the input image

fully. On the other hand, smaller image sizes result in smaller model sizes. Thus, there would be a trade-off between model size and performance.

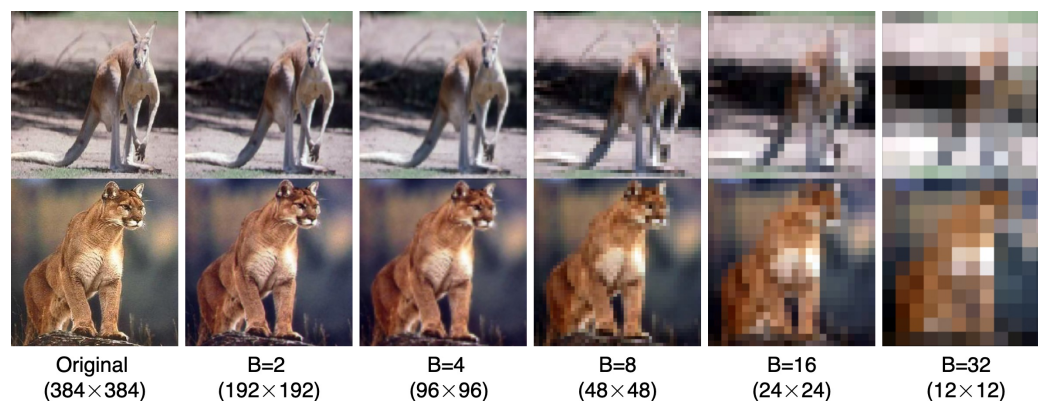


Figure 5. Visual comparison of generated feature maps named “image_0001” (above) and “image_0002” (below) from the Caltech101 dataset after changing the block size (B). We standardised the size of all feature maps to match the size of the original image for clarity.

To confirm this, we show a comparison of image classification performance with different block sizes in Table 1 using 256×256 images. It can be seen that increasing the block size reduces the input image size. Consequently, features are lost for identifying the input image, leading to a decrease in ACC.

Table 1. Image classification performance comparison on UC Merced Land Use Dataset with different block sizes under SR = 25%. The best result is labelled in **bold**.

	Block Size			
	2	4	8	16
Image Size	128×128	64×64	32×32	16×16
ACC	97.94	95.12	88.16	69.92
AUC	99.96	99.86	98.35	96.87

From these results, we can infer that image size affects model size and task accuracy. Reducing the model size helps to solve existing CL problems, and since input images’ features appear adequately preserved, and the model demonstrates sufficient classification performance even with $B = 4$, we set the block size B as 4 for subsequent experiments.

4.4.2. Effectiveness of Loss Function

Table 2 presents the results of ACC and AUC on the UC Merced Land Use Dataset, with SR = 25%, when various loss functions are combined in comparison with the widely used cross-entropy loss $\mathcal{L}$, which is defined as follows:

$$\mathcal{L} = - \sum_{k=1}^K y_k \log p_k \quad (9)$$

We observed that there is little difference in ACC and AUC between $\mathcal{L}$ and the cross-entropy of Bayesian risk $\mathcal{L}_{ce}$, which is often used in EDL. Furthermore, it is evident that adding the proposed KL divergence improves ACC and AUC. This is likely because it allocates more evidence to the target class and reduces evidence for incorrect classes, making recognition of the target class more pronounced. In addition to these loss functions, imposing an orthogonal constraint on the sampling matrix, i.e., $\mathcal{L}_{all}$, further improves the performance.

Table 2. The effectiveness of the proposed loss function on the UC Merced Land Use Dataset under SR = 25%. The best results are labelled in **bold**.

$\mathcal{L}$	$\mathcal{L}_{ce}$	$\mathcal{L}_{kl}$	$\mathcal{L}_{orth}$	ACC	AUC
✓				94.21	99.68
	✓			94.21	99.74
	✓	✓		94.37	99.82
	✓		✓	94.76	99.82
	✓	✓	✓	95.12	99.86

In Figure 6, we show the comparison of accuracy and uncertainty between the traditional EDL loss function $\mathcal{L}_{ce}$ and the loss function with proposed constraints, i.e., $\mathcal{L}_{ce} + \lambda_t \mathcal{L}_{kl}$, when changing the number of epochs. It can be seen that, shortly after the start of training, the uncertainty of $\mathcal{L}_{ce}$ becomes almost zero. On the other hand, with $\mathcal{L}_{ce} + \lambda_t \mathcal{L}_{kl}$, uncertainty decreases as training progresses, preventing it from becoming extremely small. This behavior indicates that the proposed KL-augmented loss inherits the stable convergence property of KL divergence, which has been empirically observed across various machine learning tasks.

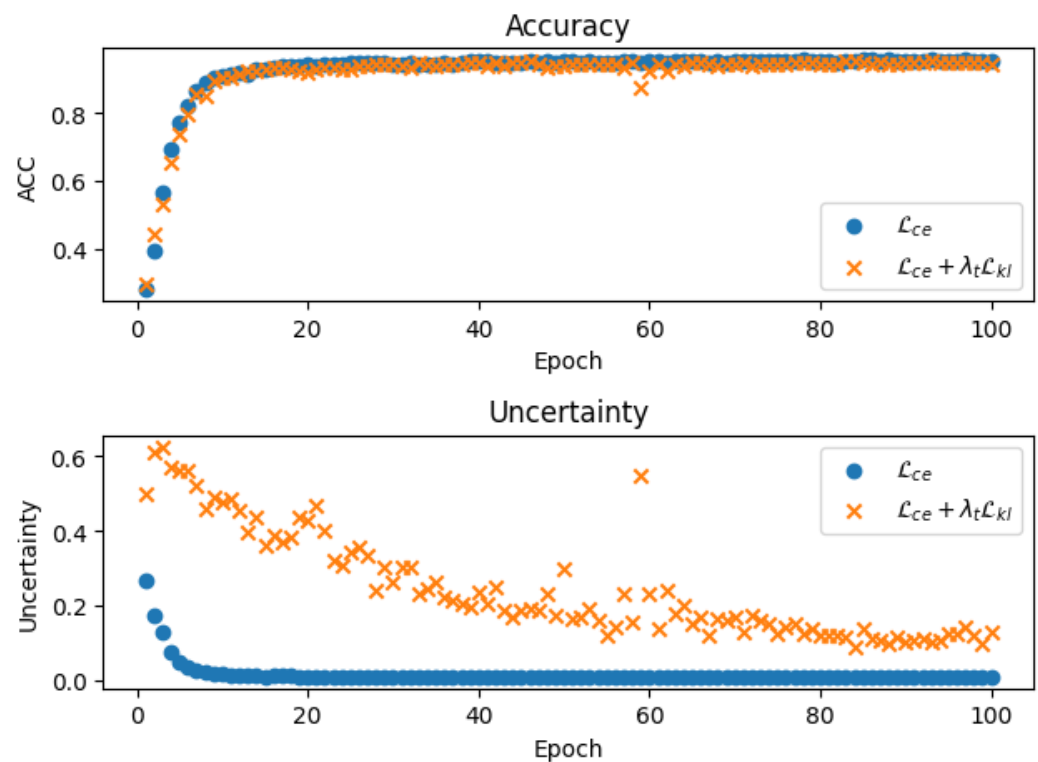


Figure 6. Accuracy and uncertainty comparison when changing the number of epochs on the Caltech101 dataset.

4.5. Comparison with State-of-the-Art CL Works

4.5.1. Configuration of Models

First, Table 3 presents the dimensions of the measurement (configuration) obtained via sampling for each model. The configurations for different SRs are listed based on an input image size of $384 \times 384 \times 3$. Both TransCL and MCS-TCL adopt a vector-based sampling approach, where computations are performed separately for each RGB channel. In contrast, MCL adopts a tensor-based sampling method, resulting in three-dimensional measurements. To ensure a fair comparison, we adjust the configuration values of MCL

to closely match those of the vector-based SR. Please note that, after sampling, MCL and TransCL expand the image dimensions to the original input size.

Table 3. Different measurement configurations between the tensor (MCL) and vector (TransCL and MCS-TCL). The configurations represent the dimensions of the measurement obtained for the corresponding SR when the input image size is $384 \times 384 \times 3$.

Model Name	Configuration	SR
MCL	$334 \times 334 \times 1$	0.2522
TransCL	$36,864 \times 3$	0.2500
MCS-TCL (ours)	$36,864 \times 3$	0.2500
MCL	$235 \times 235 \times 1$	0.1248
TransCL	$18,432 \times 3$	0.1250
MCS-TCL (ours)	$18,432 \times 3$	0.1250
MCL	$163 \times 163 \times 1$	0.0601
TransCL	9216×3	0.0625
MCS-TCL (ours)	9216×3	0.0625

4.5.2. Qualitative Comparison of Input Image to Task-Specified Network

We input images into the task-specific network in each CL framework, as shown in Figure 7. To quantitatively assess the image quality, we use PSNR (dB) and SSIM [67]. PSNR measures pixel-wise errors between images, while SSIM assesses perceptual quality by using low-dimensional features, such as contrast. Higher values for both metrics indicate better image quality. Since MCL and TransCL interpolate the measurement to match the input image size, they can compute the PSNR and SSIM. In contrast, our method does not inherently perform such interpolation. Therefore, we align the generated feature maps with the input image size, and the results are presented in Figure 7.

The objects in the images input by MCL are blurry, and colour is lacking. TransCL suffers from boundary lines and noise due to block-based compressive sensing (BCS) [56], and its impact becomes more pronounced as the SR decreases. Therefore, the PSNR and SSIM of these images are extremely small, especially for TransCL. In contrast, through our approach, there is little to no visual change even as the SR decreases. Moreover, considering the high PSNR and SSIM, it seems that the essential features required for identifying the input images are retained.

4.5.3. Model Size Comparison

Table 4 shows the number of measurements (Meas), model size (MB), multi-adds (G), and number of parameters for each framework when receiving a particular image. MCL and TransCL restore the measurements to the same dimensions as the input image by using HOSVD [68] and CS reconstruction, respectively. Therefore, the number of measurements to be input into the task-specific network in MCL and TransCL is $384 \times 384 \times 3 = 442,368$ regardless of SR, resulting in large model sizes, multi-adds, and numbers of parameters. On the other hand, MCS-TCL generates M feature maps of size $96 \times 96 \times 3$. Since the input is divided into M times, its size remains fixed akin to the batch size (96, 96, 3). Therefore, the model size, multi-adds, and number of parameters remain unchanged due to SR, with only the number of input feature maps M varying. We can see that MCS-TCL has the smallest model size, fewest multi-adds, and lowest parameter count at all SRs. When SR is 0.625, our model achieves a reduction in model size of 84.98%, a reduction in multi-adds of 49.94%, and a reduction in parameter number of 79.33% compared to existing works.

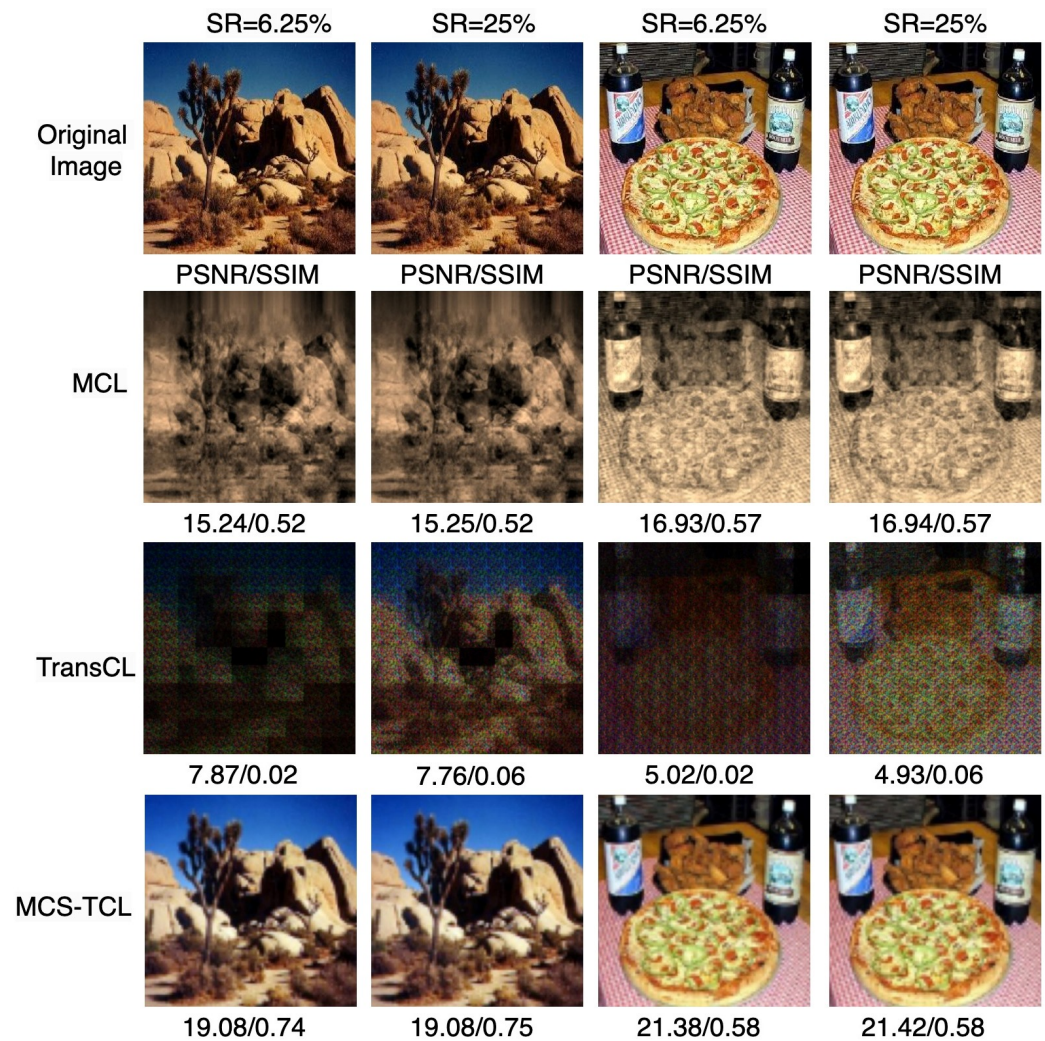


Figure 7. Visual comparison of the images inputted into the task-specific network in each CL framework.

Table 4. The number of measurements (Meas), model size, multi-adds and the number of parameters are reported when inputting the $384 \times 384 \times 3$ image to each framework by changing the SR. The best result is labelled in **bold**.

SR	Method	Meas	Model Size (MB)	Multi-Adds (G)	Number of Params
6.25%	MCL [23]	442,368	16,650.11	54.4	423,250,378
	TransCL [24]	442,368	17,556.83	54.4	423,437,770
	MCS-TCL	27,648	2500.32	27.23	87,502,181
12.5%	MCL [23]	442,368	16,650.11	54.4	423,250,378
	TransCL [24]	442,368	17,557.66	54.4	423,643,594
	MCS-TCL	55,296	2500.32	27.23	87,502,181
25%	MCL [23]	442,368	16,650.11	54.4	423,250,378
	TransCL [24]	442,368	17,559.23	54.4	424,036,810
	MCS-TCL	110,592	2500.32	27.23	87,502,181

4.5.4. Classification Accuracy Comparison

In this subsection, we present the comparison of the image classification performance as shown in Table 5. It can be seen that although MCS-TCL conducts inference from few measurements, it performs as well as or better than MCL and TransCL, achieving excellent performance in the image classification tasks. We report an improvement of up to 3.34% in ACC and 0.71% in AUC on the UC Merced Land Use Dataset under SR = 6.25% compared

with TransCL. The relatively lower performance on RESISC45 would arise from the inherent characteristics of remote sensing images, which exhibit large intra-class variations and ambiguous class boundaries, making classification more challenging. Nevertheless, the performance drop is modest (only 1–3% compared to Caltech101 and UC Merced under any SR), and our framework still achieves approximately 93% accuracy, demonstrating competitive performance even on this challenging dataset.

Table 5. Image classification performance comparison on Caltech101, UC Merced Land Use Dataset and RESISC45 dataset with different SRs. Accuracy and AUC are presented below. The best result is labelled in **bold**.

Dataset	Method	SR		
		6.25%	12.5%	25%
Caltech101	MCL [23]	90.67/87.88	90.02/88.42	91.05/87.52
	TransCL [24]	94.24/89.45	94.28/ 90.53	94.47/ 90.49
	MCS-TCL	94.09/88.53	94.66/88.54	94.58/88.54
UC Merced	MCL [23]	84.60/98.97	86.67/99.27	86.19/99.12
	TransCL [24]	93.49/99.21	93.33/99.90	95.24/99.44
	MCS-TCL	96.83/99.92	96.43/99.93	96.71/99.92
RESISC45	MCL [23]	80.61/98.84	80.74/99.03	80.78/98.87
	TransCL [24]	90.78/99.45	91.53/99.54	91.88/99.42
	MCS-TCL	93.24/99.75	93.19/99.75	93.15/99.75

4.5.5. Robustness to Degraded Input Images

Images input into a CL system may not always be clean. In real-world applications, they often contain noise or even appear blurred. We demonstrate that our proposed method remains robust even under such degraded image conditions. We apply a Gaussian filter with horizontal and vertical blurring strengths of 10 and 20, respectively. The larger the kernel size, the more blurred the image borders become. We present the comparison results of class probabilities when inputting images with these parameters changed to 0 (original), 13 and 25, as shown in Figure 8. Since MCS-TCL follows EDL, the uncertainty of predictions is also quantified. For the original image, all methods estimate the correct class with a high probability. However, when the kernel size is increased, all methods often output incorrect classes. The issue is that existing CL methods output incorrect answers with high probabilities, especially TransCL, making the system unreliable for users. On the other hand, MCS-TCL outputs relatively few incorrect classes with very small probabilities. Since MCS-TCL is trained with feature maps that lack detailed information on the input image, as shown in Figure 7, our proposed method would be robust to noise. In other words, it excels at recognising global features, and the inference results may not be affected by changes in detailed information.

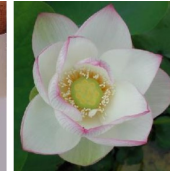
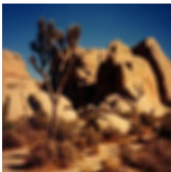
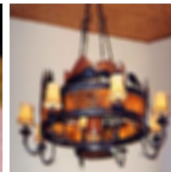
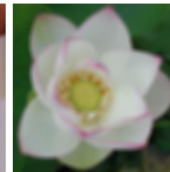
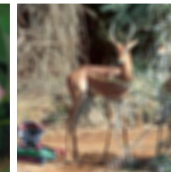
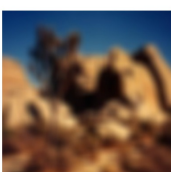
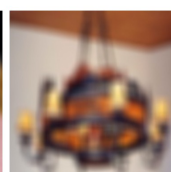
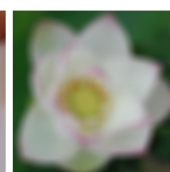
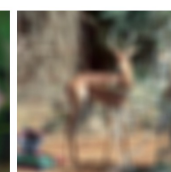
Original Image					
	MCL 25.45/- TransCL 99.77/- MCS-TCL 82.82/15.73	MCL 97.16/- TransCL 91.02/- MCS-TCL 80.84/18.05	MCL 63.58/- TransCL 94.85/- MCS-TCL 78.75/19.96	MCL 90.34/- TransCL 94.36/- MCS-TCL 82.16/16.69	MCL 80.78/- TransCL 96.94/- MCS-TCL 86.80/11.74
Blurred Image (Kernel size = 13)					
	MCL 8.84 /- TransCL 77.48 /- MCS-TCL 66.19/32.39	MCL 12.58 /- TransCL 55.92 /- MCS-TCL 1.13 /66.27	MCL 65.81/- TransCL 42.24/- MCS-TCL 62.23/36.29	MCL 55.95/- TransCL 85.43/- MCS-TCL 77.77/21.57	MCL 5.88 /- TransCL 21.26 /- MCS-TCL 61.92/35.15
Blurred Image (Kernel size = 25)					
	MCL 6.31 /- TransCL 15.79 /- MCS-TCL 45.32/52.60	MCL 24.09 /- TransCL 37.69 /- MCS-TCL 1.28 /76.73	MCL 24.68/- TransCL 12.83 /- MCS-TCL 1.46/36.29	MCL 17.89 /- TransCL 22.46 /- MCS-TCL 83.78/15.63	MCL 7.53 /- TransCL 13.39 /- MCS-TCL 1.20 /76.26

Figure 8. Comparison of class probability when changing the kernel size of the Gaussian filter under SR = 25% on the Caltech101 dataset. Our framework also quantifies uncertainty. Results that estimated incorrect classes are indicated in blue.

5. Conclusions

In this paper, we introduced Multi-functional Compressive Sensing-based Trustworthy Compressive Learning (MCS-TCL), a novel framework that simultaneously performs image sampling, compression, and feature extraction. By leveraging the structural similarity between CS sampling and convolution operations, we demonstrated that CS sampling can be reformulated as a convolutional process without violating its theoretical principles. Additionally, we proposed a modified EDL loss function that improves uncertainty quantification by suppressing spurious evidence assigned to incorrect classes and reinforcing confidence in the target class. Extensive experimental evaluations confirmed that MCS-TCL achieves state-of-the-art performance in both predictive accuracy and computational efficiency, highlighting its practical applicability to real-world vision tasks.

Author Contributions: Conceptualization, F.K.; methodology, F.K.; software, F.K.; validation, F.K.; investigation, F.K.; data curation, F.K.; writing—original draft preparation, F.K.; visualization, F.K. and J.Y.; project administration, F.K.; writing—review and editing, J.Y. and J.Z.; supervision, J.Z.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by JSPS KAKENHI, grant number JP22K12101.

Data Availability Statement: The datasets used in this study are publicly available: Caltech101 is available at <https://data.caltech.edu/records/mzrjq-6wc02> (accessed on 4 September 2025); UC Merced Land Use Dataset is available at <http://weege.vision.ucmerced.edu/datasets/landuse.html> (accessed on 4 September 2025); and NWPU-RESISC45 is available at https://1drv.ms/u/s!AmgKYzARBl5ca3HNaHilzp_IXjs (accessed on 4 September 2025). The source code and result data are available via the following GitHub (version v1.0.0) repository: <https://github.com/fuma8/Trustworthy-CL> (accessed on 4 September 2025).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Candès, E.J.; Wakin, M.B. An introduction to compressive sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 21–30. [CrossRef]
2. Liutkus, A.; Martina, D.; Popoff, S.; Chardon, G.; Katz, O.; Lerosey, G.; Gigan, S.; Daudet, L.; Carron, I. Imaging with nature: Compressive imaging using a multiply scattering medium. *Sci. Rep.* **2014**, *4*, 5552. [CrossRef]
3. Lustig, M.; Donoho, D.L.; Santos, J.M.; Pauly, J.M. Compressed sensing MRI. *IEEE Signal Process. Mag.* **2008**, *25*, 72–82. [CrossRef]
4. Duarte, M.F.; Davenport, M.A.; Takhar, D.; Laska, J.N.; Sun, T.; Kelly, K.F.; Baraniuk, R.G. Single-pixel imaging via compressive sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 83–91. [CrossRef]
5. Rousset, F.; Ducros, N.; Farina, A.; Valentini, G.; d’Andrea, C.; Peyrin, F. Adaptive basis scan by wavelet prediction for single-pixel imaging. *IEEE Trans. Comput. Imaging* **2016**, *3*, 36–46. [CrossRef]
6. Wu, Z.; Zhang, J.; Mou, C. Dense deep unfolding network with 3D-CNN prior for snapshot compressive imaging. *arXiv* **2021**, arXiv:2109.06548. [CrossRef]
7. Dong, W.; Shi, G.; Li, X.; Ma, Y.; Huang, F. Compressive Sensing via Nonlocal Low-Rank Regularization. *IEEE Trans. Image Process.* **2014**, *23*, 3618–3632. [CrossRef]
8. Kim, Y.; Nadar, M.S.; Bilgin, A. Compressed sensing using a Gaussian scale mixtures model in wavelet domain. In Proceedings of the 2010 IEEE International Conference on Image Processing, Hong Kong, China, 26–29 September 2010; IEEE: New York, NY, USA, 2010; pp. 3365–3368.
9. Zhang, J.; Zhao, C.; Zhao, D.; Gao, W. Image compressive sensing recovery using adaptively learned sparsifying basis via L0 minimization. *Signal Process.* **2014**, *103*, 114–126. [CrossRef]
10. Zhang, J.; Zhao, D.; Gao, W. Group-based sparse representation for image restoration. *IEEE Trans. Image Process.* **2014**, *23*, 3336–3351. [CrossRef] [PubMed]
11. Zhang, J.; Zhao, C.; Gao, W. Optimization-inspired compact deep compressive sensing. *IEEE J. Sel. Top. Signal Process.* **2020**, *14*, 765–774. [CrossRef]
12. You, D.; Xie, J.; Zhang, J. ISTA-NET++: Flexible deep unfolding network for compressive sensing. In Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME), Shenzhen, China, 5–9 July 2021; IEEE: New York, NY, USA, 2021; pp. 1–6.
13. Song, J.; Chen, B.; Zhang, J. Memory-augmented deep unfolding network for compressive sensing. In Proceedings of the 29th ACM International Conference on Multimedia, Virtual, 20–24 October 2021; pp. 4249–4258.
14. Chen, B.; Zhang, J. Content-aware scalable deep compressed sensing. *IEEE Trans. Image Process.* **2022**, *31*, 5412–5426. [CrossRef]
15. Song, J.; Mou, C.; Wang, S.; Ma, S.; Zhang, J. Optimization-Inspired Cross-Attention Transformer for Compressive Sensing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 6174–6184.
16. Mohassel, P.; Zhang, Y. Secureml: A system for scalable privacy-preserving machine learning. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; IEEE: New York, NY, USA, 2017; pp. 19–38.
17. Calderbank, R.; Jafarpour, S.; Schapire, R. Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain. *Preprint* **2009**. Available online: <https://www.semanticscholar.org/paper/Compressed-Learning-%3A-Universal-Sparse-Reduction-in-Calderbank/627c14fe9097d459b8fd47e8a901694198be9d5d#citing-papers> (accessed on 4 September 2025).
18. Reboredo, H.; Renna, F.; Calderbank, R.; Rodrigues, M.R. Compressive classification. In Proceedings of the 2013 IEEE International Symposium on Information Theory, Istanbul, Turkey, 7–12 July 2013; IEEE: New York, NY, USA, 2013; pp. 674–678.
19. Reboredo, H.; Renna, F.; Calderbank, R.; Rodrigues, M.R.D. Projections designs for compressive classification. In Proceedings of the 2013 IEEE Global Conference on Signal and Information Processing, Austin, TX, USA, 3–5 December 2013; pp. 1029–1032. [CrossRef]
20. Lohit, S.; Kulkarni, K.; Turaga, P.; Wang, J.; Sankaranarayanan, A.C. Reconstruction-free inference on compressive measurements. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 June 2015; pp. 16–24.
21. Adler, A.; Elad, M.; Zibulevsky, M. Compressed learning: A deep neural network approach. *arXiv* **2016**, arXiv:1610.09615. [CrossRef]
22. Zisselman, E.; Adler, A.; Elad, M. Compressed learning for image classification: A deep neural network approach. In *Handbook of Numerical Analysis*; Elsevier: Amsterdam, The Netherlands, 2018; Volume 19, pp. 3–17.
23. Tran, D.T.; Yamaç, M.; Degerli, A.; Gabbouj, M.; Iosifidis, A. Multilinear compressive learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *32*, 1512–1524. [CrossRef]

24. Mou, C.; Zhang, J. TransCL: Transformer makes strong and flexible compressive learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 5236–5251. [CrossRef] [PubMed]
25. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
26. Sensoy, M.; Kaplan, L.; Kandemir, M. Evidential deep learning to quantify classification uncertainty. *Adv. Neural Inf. Process. Syst.* **2018**, *31*. [CrossRef]
27. Kimishima, F. Multi-functional Compressive Sensing Sampling-based Trustworthy Compressive Learning. In Proceedings of the 2025 Data Compression Conference (DCC), Snowbird, UT, USA, 18–21 March 2025.
28. Chen, S.S.; Donoho, D.L.; Saunders, M.A. Atomic decomposition by basis pursuit. *SIAM Rev.* **2001**, *43*, 129–159. [CrossRef]
29. Zhang, J.; Ghanem, B. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1828–1837.
30. Shi, W.; Jiang, F.; Liu, S.; Zhao, D. Image compressed sensing using convolutional neural network. *IEEE Trans. Image Process.* **2019**, *29*, 375–388. [CrossRef]
31. Canh, T.N.; Jeon, B. Multi-scale deep compressive sensing network. In Proceedings of the 2018 IEEE Visual Communications and Image Processing (VCIP), Taichung, Taiwan, 9–12 December 2018; IEEE: New York, NY, USA, 2018; pp. 1–4.
32. Wu, Y.; Rosca, M.; Lillicrap, T. Deep compressed sensing. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; PMLR: San Diego, CA, USA, 2019; pp. 6850–6860.
33. Shi, W.; Jiang, F.; Liu, S.; Zhao, D. Scalable convolutional neural network for image compressed sensing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12290–12299.
34. Zhang, Z.; Liu, Y.; Liu, J.; Wen, F.; Zhu, C. AMP-Net: Denoising-based deep unfolding for compressive image sensing. *IEEE Trans. Image Process.* **2020**, *30*, 1487–1500. [CrossRef]
35. Davenport, M.A.; Duarte, M.F.; Wakin, M.B.; Laska, J.N.; Takhar, D.; Kelly, K.F.; Baraniuk, R.G. The smashed filter for compressive classification and target recognition. In Proceedings of the Computational Imaging V, San Jose, CA, USA, 28 February 2007; SPIE: St. Bellingham, WA, USA, 2007; Volume 6498, pp. 142–153.
36. Baraniuk, R.G.; Wakin, M.B. Random projections of smooth manifolds. *Found. Comput. Math.* **2009**, *9*, 51–77. [CrossRef]
37. Kulkarni, K.; Turaga, P. Recurrence textures for human activity recognition from compressive cameras. In Proceedings of the 2012 19th IEEE International Conference on Image Processing, Orlando, FL, USA, 30 September–3 October 2012; IEEE: New York, NY, USA, 2012; pp. 1417–1420.
38. Kulkarni, K.; Turaga, P. Reconstruction-free action inference from compressive imagers. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 772–784. [CrossRef]
39. Calderbank, R.; Jafarpour, S. Finding needles in compressed haystacks. In Proceedings of the 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Kyoto, Japan, 25–30 March 2012; IEEE: New York, NY, USA, 2012; pp. 3441–3444.
40. Wimalajeewa, T.; Chen, H.; Varshney, P.K. Performance limits of compressive sensing-based signal classification. *IEEE Trans. Signal Process.* **2012**, *60*, 2758–2770. [CrossRef]
41. Haupt, J.; Castro, R.; Nowak, R.; Fudge, G.; Yeh, A. Compressive Sampling for Signal Classification. In Proceedings of the 2006 Fortieth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 29 October–1 November 2006; pp. 1430–1434. [CrossRef]
42. Lohit, S.; Kulkarni, K.; Turaga, P. Direct inference on compressive measurements using convolutional neural networks. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 1913–1917. [CrossRef]
43. Hollis, B.; Patterson, S.; Trinkle, J. Compressed Learning for Tactile Object Recognition. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1616–1623. [CrossRef]
44. Xu, Y.; Kelly, K.F. Compressed domain image classification using a multi-rate neural network. *arXiv* **2019**, arXiv:1901.09983.
45. Gal, Y. Uncertainty in Deep Learning. Ph.D. Dissertation, Cambridge University, Cambridge, UK, 2016. Available online: https://scholar.google.com/citations?view_op=view_citation&hl=ja&user=SlayDoQAAAAJ&scstart=300&pagesize=100&sortby=pubdate&citation_for_view=SlayDoQAAAAJ:kNdYIx-mwKoC (accessed on 4 September 2025).
46. Guo, C.; Pleiss, G.; Sun, Y.; Weinberger, K.Q. On calibration of modern neural networks. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; PMLR: San Diego, CA, USA, 2017; pp. 1321–1330.
47. Malinin, A.; Gales, M. Predictive uncertainty estimation via prior networks. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 7047–7058.
48. Dempster, A.P. Upper and lower probabilities induced by a multivalued mapping. In *Classic Works of the Dempster-Shafer Theory of Belief Functions*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 57–72.
49. Dempster, A.P. A generalization of Bayesian inference. *J. R. Stat. Soc. Ser. B (Methodol.)* **1968**, *30*, 205–232. [CrossRef]

50. Gao, J.; Chen, M.; Xu, C. Collecting Cross-Modal Presence-Absence Evidence for Weakly-Supervised Audio-Visual Event Perception. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 18827–18836.
51. Li, B.; Han, Z.; Li, H.; Fu, H.; Zhang, C. Trustworthy long-tailed classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 6970–6979.
52. Xu, Z.; Yue, X.; Lv, Y.; Liu, W.; Li, Z. Trusted fine-grained image classification through hierarchical evidence fusion. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; Volume 37, pp. 10657–10665.
53. Han, Z.; Zhang, C.; Fu, H.; Zhou, J.T. Trusted multi-view classification with dynamic evidential fusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 2551–2566. [[CrossRef](#)] [[PubMed](#)]
54. Ren, J.; Jiang, L.; Peng, H.; Liu, Z.; Wu, J.; Philip, S.Y. Evidential temporal-aware graph-based social event detection via dempster-shafer theory. In Proceedings of the 2022 IEEE International Conference on Web Services (ICWS), Barcelona, Spain, 10–16 July 2022; IEEE: New York, NY, USA, 2022; pp. 331–336.
55. Ren, J.; Peng, H.; Jiang, L.; Liu, Z.; Wu, J.; Yu, Z.; Philip, S.Y. Uncertainty-guided boundary learning for imbalanced social event detection. *IEEE Trans. Knowl. Data Eng.* **2023**, *36*, 2701–2715. [[CrossRef](#)]
56. Gan, L. Block compressed sensing of natural images. In Proceedings of the 2007 15th International Conference on Digital Signal Processing, Cardiff, UK, 1–4 July 2007; IEEE: New York, NY, USA, 2007; pp. 403–406.
57. Gao, X.; Zhang, J.; Che, W.; Fan, X.; Zhao, D. Block-based compressive sensing coding of natural images by local structural measurement matrix. In Proceedings of the 2015 Data Compression Conference, Snowbird, UT, USA, 7–9 April 2015; IEEE: New York, NY, USA, 2015; pp. 133–142.
58. Burrus, C.S.; Parks, T. *Convolution Algorithms*; Citeseer: New York, NY, USA, 1985; Volume 6, p. 15.
59. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [[CrossRef](#)]
60. Krizhevsky, A.; Hinton, G. *Learning Multiple Layers of Features from Tiny Images*; Technical Report; University of Toronto: Toronto, ON, Canada, 2009. Available online: https://scholar.google.com/citations?view_op=view_citation&hl=ja&user=xegzhJcAAAAJ&citation_for_view=xegzhJcAAAAJ:d1gkVwhDpl0C (accessed on 4 September 2025).
61. Li, F.F.; Andreeto, M.; Ranzato, M.; Perona, P. Caltech 101. CaltechDATA. 2022. Available online: <https://data.caltech.edu/records/mzrqj-6wc02> (accessed on 4 September 2025).
62. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
63. Cheng, G.; Han, J.; Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proc. IEEE* **2017**, *105*, 1865–1883. [[CrossRef](#)]
64. McClish, D.K. Analyzing a portion of the ROC curve. *Med. Decis. Mak.* **1989**, *9*, 190–195. [[CrossRef](#)]
65. Loshchilov, I.; Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv* **2016**, arXiv:1608.03983.
66. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 721.
67. Wang, Z.; Bovik, A.; Sheikh, H.; Simoncelli, E. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
68. De Lathauwer, L.; De Moor, B.; Vandewalle, J. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.* **2000**, *21*, 1253–1278. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.