

# MetaMorphosis: Task-oriented Privacy Cognizant Feature Generation for Multi-task Learning

Md Adnan Arefeen  
aa4cy@mail.umkc.edu

University Of Missouri-Kansas City  
Kansas City, Missouri, USA

Md Yusuf Sarwar Uddin  
muddin@umkc.edu

University Of Missouri-Kansas City  
Kansas City, Missouri, USA

Zhouyu Li  
zli85@ncsu.edu

North Carolina State University  
Raleigh, North Carolina, USA

Anupam Das

anupam.das@ncsu.edu  
North Carolina State University  
Raleigh, North Carolina, USA

## ABSTRACT

With the growth of computer vision applications, deep learning, and edge computing contribute to ensuring practical collaborative intelligence (CI) by distributing the workload among edge devices and the cloud. However, running separate single-task models on edge devices is inefficient regarding the required computational resource and time. In this context, *multi-task learning* allows leveraging a single deep learning model for performing multiple tasks, such as semantic segmentation and depth estimation on incoming video frames. This single processing pipeline generates common *deep features* that are shared among multi-task modules. However, in a collaborative intelligence scenario, generating common deep features has two major issues. First, the deep features may inadvertently contain input information exposed to the downstream modules (violating *input privacy*). Second, the generated universal features expose a piece of collective information than what is intended for a certain task, in which features for one task can be utilized to perform another task (violating *task privacy*). This paper proposes a novel deep learning-based privacy-cognizant feature generation process called “MetaMorphosis” that limits inference capability to specific tasks at hand. To achieve this, we propose a channel *squeeze-excitation* based feature metamorphosis module, *Cross-SEC*, to achieve distinct attention of all tasks and a de-correlation loss function with *differential-privacy* to train a deep learning model that produces distinct privacy-aware features as an output for the respective tasks. With extensive experimentation on four datasets consisting of diverse images related to scene understanding and facial attributes, we show that MetaMorphosis outperforms recent adversarial learning and universal feature generation methods by guaranteeing privacy requirements in an efficient way for image and video analytics.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*IoTDDI '23, May 09–12, 2023, San Antonio, TX, USA*

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
ACM ISBN 979-8-4007-0037-8/23/05...\$15.00  
<https://doi.org/10.1145/3576842.3582372>

## CCS CONCEPTS

• **Computing methodologies** → **Computer vision tasks; Scene understanding**; • **Security and privacy** → **Privacy protections**.

## KEYWORDS

Multi-task learning, neural networks, collaborative intelligence, differential privacy, task privacy

## ACM Reference Format:

Md Adnan Arefeen, Zhouyu Li, Md Yusuf Sarwar Uddin, and Anupam Das. 2023. MetaMorphosis: Task-oriented Privacy Cognizant Feature Generation for Multi-task Learning. In *International Conference on Internet-of-Things Design and Implementation (IoTDDI '23), May 09–12, 2023, San Antonio, TX, USA*. ACM, New York, NY, USA, 13 pages. <https://doi.org/10.1145/3576842.3582372>

## 1 INTRODUCTION

Computer vision-based technologies have seen widespread adoption over recent years due to improved performance. This use is not limited to the rapid adoption of facial recognition technology but extends to autonomous driving [37], scene recognition, and more [9, 29]. As a result, organizations and even cities have started utilizing video feeds to carry out various automated tasks. However, while computer vision-based technologies provide new opportunities, they also raise privacy concerns and call for novel solutions to ensure adequate privacy protection.

One trivial way to protect sensitive information is not to send protected information outside the organization by any means, i.e., to train the deep-learning model within the respective organization providing the inputs. That implies the input-providing organizations (i.e., *producers/publishers/feature providers*) also have to construct various models for different tasks, e.g., object detection, depth estimation, etc. One of the drawbacks of this approach is that organizations owning the video/input feed will also need to develop the entire analytic pipeline whose primary interest may be orthogonal to building deep learning models, such as hospitals or grocery stores. Organizations can also resort to *video analytics as a service* where companies are now offering essential video processing pipelines as paid services [13, 27]. However, outsourcing video feeds to cloud services also raises privacy concerns as video feeds can be used to infer various sensitive information. As an alternative, a hybrid approach can also be adopted where instead of sending the raw input, some useful derived features are shared with the third

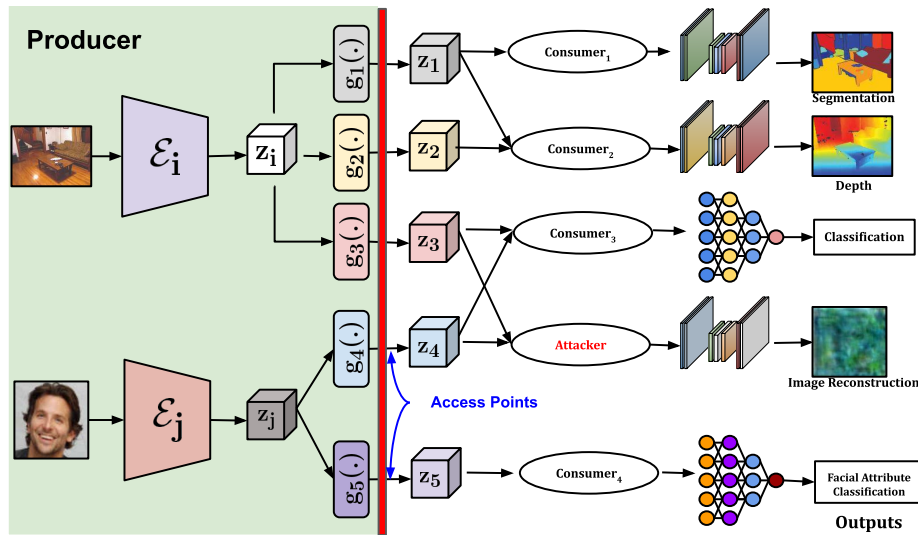


Figure 1: Overview of MetaMorphosis.

party (i.e., *consumers*) to prevent unintended information leakage. This approach is known as *collaborative intelligence*.

In collaborative intelligence, intelligence is shared across more than one entity to split the computation overload, where one entity can run a portion of a deep model and send the intermediate partial output as “features” to another entity for further computation. In this way, the input can be replaced by meaningful features. One of the popular architectures adopted in this context is the *multi-task learning* paradigm, which offers an efficient solution to reduce computational resources across different analytic tasks. The efficiency comes through the introduction of a shared deep layer to produce *universal* features usable by all downstream tasks [21]. Unfortunately, it does not fully diminish the privacy concern. The shared features, also called intermediate representations, can be reverted to the actual input, thus violating *input privacy* and affecting the whole notion of providing deep features rather than the input itself. Similarly, the universal features generated for multi-task learning, when subscribed by different downstream tasks, can also leak unintended information leading to violating *task privacy*. From a privacy and business perspective, if the task-oriented features differ, the producers can offer specific features based on consumers’ objectives and hide private attribute information from each task. For example, for a given image, the feature for segmentation will be different than the feature for depth estimation or classification.

In this paper, we focus on a publisher-subscriber-based multi-party communication system where one party acts as a publisher, and the rest acts as a consumer/subscriber (Figure 1). The feature publishing party also known as the feature provider or the publisher holds the data and private information, and with proper intelligent tweaking, it provides privacy-aware task-variant features to consumers. Consumers, on the other hand, consume the features rather than the input and train a task according to the features. Without loss of generality, we can assume that the producer can securely share the output with the consumer so that the consumer can train the rest of the remaining part using the task-variant features. With

respect to deep learning tasks, mapping the feature to output does not violate our assumption of securing object attributes rather than the type of objects for classification cases. In the case of object detection, segmentation, or depth estimation, it can be shared to blur the attribute of the objects.

To achieve data and task privacy, we propose MetaMorphosis, which consists of two modules, (a) a *private encoder* trained using differential privacy, and (b) a *task metamorphosis* module for each task for task privacy. The private encoder protects identifiable information from input, which we refer to as input obfuscation. The task metamorphosis modules help to form *distinct* features for each task. The privacy of the encoder depends on the requirement of privacy based on the data. So, the producer can hold private and non-private encoder submodules to offer both options to the consumers based on the privacy requirement. The whole functionality of the producer will be obscured so that the consumer cannot determine the types and characteristics of the construction of the producers.

Several challenges arise when offering task-oriented privacy-aware features. Firstly, the joint training of input obfuscation and task privacy in a single phase makes the whole process challenging due to the uncertainty of leaking unintended information to task-specific features. Secondly, a sophisticated feature morphosis module is required to achieve the right balance of performance and privacy. Finally, the proposed approach has to be scalable to facilitate new tasks with minimal training effort. In order to address the challenges, we propose MetaMorphosis and specify the contribution of this work as follows.

- We propose MetaMorphosis, which ensures input obfuscation and privacy-aware task variant feature generation to prevent information leakage through the shared features while still providing acceptable outcomes for the intended tasks.
- We propose a novel task metamorphosis module *Cross-SEC* that maintains or even improves the performance in addition to producing distinct task-specific features.

**Table 1: Comparison of MetaMorphosis with recent literature**

Characteristics	MetaMorphosis	DeepObfuscator [18]	TIPRDC [17]	ALPPTOR [42]	P-FEAT [10]
Input obfuscation	✓	✓	✓	✓	✓
Noisy models parameters	✓	✗	✗	✗	✗
Task privacy	✓	✗	✗	✗	✗
Scalable	Good	Poor	Poor	Poor	Poor
Quantifiable privacy	✓	✗	✓	✗	✗
Feature sharing	Task-specific	Universal	Task-specific	Task-specific	Universal
Training budget	LOW	HIGH	HIGH	HIGH	HIGH

- We reduce the training time of task-specific feature generation by collaborating on the task-invariant and metamorphosis modules.
- The scope for sequential and parallel training helps MetaMorphosis improve scalability compared to recent adversarial learning methods, such as [18].

The rest of the paper is organized as follows. Section 2 describes the motivation of our work by comparing it to similar works. Section 3 explains the MetaMorphosis. We present the findings in Section 4. Section 4.1 defines the datasets and metrics that we use in the analysis. We also evaluate training and inference results after deployment of MetaMorphosis in Section 5. An ablation study is conducted to show the reasons behind choosing specific modules and parameters in Section 6. Related work is added in Section 7. Finally, we conclude in Section 8.

## 2 MOTIVATION AND CHALLENGES

Various kinds of deep learning models [4, 5, 11, 33, 40] have been proposed to resolve visual applications with multi-task learning setups such as semantic segmentation, and depth estimation, efficiently. Khattar et al. [16] propose a multi-task learning framework where domain-agnostic features are learned to improve the model performance on both object detection and saliency prediction tasks with limited data. Meanwhile, techniques such as knowledge distillation fit well with multi-task training where knowledge is distilled from single models by minimizing the distance, thus contributing to fast training of multi-task models [19–21]. As a universal feature is shared for all downstream tasks, it is computationally efficient but raises a privacy concern while sharing with outside agents due to offering a common feature for all tasks. Similar behavior patterns can be found in other recent literature [2, 3] where features from multiple layers of deep models are used to form the universal features and image classification task is accomplished.

To preserve the privacy of the universal features, several adversarial learning algorithms have been proposed [17, 18] to obfuscate intermediate representation. In this, adversarial decoders and classifiers are trained jointly with the intended classification task to obfuscate features [18]. TIPRDC [17] is also designed to hide private information from the feature vector while retaining the feature’s utility regarding the primary task through a hybrid training algorithm. ALPPTOR [42] framework proposed a GAN loss to prevent model-inversion attack by adversarial reconstruction learning and provide task-oriented representations for binary classification tasks. P-FEAT [10] proposed two adversarial objectives for privacy-preserving feature encoding-based adversarial training,

which considers privacy attributes and privacy-attribute agnostic scenarios. In split federated learning [35], intermediate features of IID data are shared with the server and the server returns the gradients back to clients.

These methods face drawbacks at the time of adding a new task to the framework, as adversary decoders and classifiers need to be trained again with the addition of new tasks. Another disadvantage is the need for ground truth in all tasks to train the whole pipeline to prevent features from being attacked by intruders. With the development of edge computing technology, the emerging collaborative intelligent technique allows computational-constrained devices to participate as end-users where sharing of intermediate representation takes the first place to connect two entities. Table 1 compares the overview of MetaMorphosis with related recent literature to show the effectiveness of MetaMorphosis. Rather than training a decoder to decode the intermediate representation to a noise, MetaMorphosis uses differential privacy along with an intelligent split learning method, which can guarantee obfuscation of input as well as achieve target performance. MetaMorphosis assures *task-privacy* by making the intermediate features distinct from each other, which limits the necessity to have ground truth for all tasks. With ground truth, extra DNN models are required for training an adversary classifier. At the time of addition of a new task, MetaMorphosis learns to make the new task features distinct from the already added tasks. Thus, MetaMorphosis ensures better scalability and a low training budget. To produce the distinct features, MetaMorphosis offers a novel metamorphosis module. In summary, MetaMorphosis answers the following key questions:

- How to reduce the input information leak while sending deep features rather than the input itself?
- How to overcome privacy issues regarding universal features for all tasks?
- How to design a lightweight task metamorphosis module so that the performance drop should be negligible and almost similar to the performance of a single task?

## 3 METAMORPHOSIS

Generating features for different tasks is the core part of MetaMorphosis and as a result, several considerations are undertaken in the construction of the producer to enhance the target performance and privacy, reduce memory issues, and latency of the system.

### 3.1 Privacy Cognizant Feature Generation

At first, task-oriented single models cannot be provided due to zero task privacy for independent training. In addition, memory requirements will increase when new tasks are assigned. So, a multi-task model is required to reduce the number of models. In this way, one model can provide a universal encoder to produce the features for all tasks. But the drawback of the latter method for producing universal representation lies in the degradation of performance and privacy in some cases for de-correlated tasks. For example, the data owner can issue a restriction on reconstructing the images from the encoded features for facial image attribute classification. But for semantic segmentation or depth estimation tasks, privacy can be imposed on the feature generation so that unique features are generated for each task at hand. A universal representation fails to either provide high accuracy for all tasks or prevent privacy attacks due to providing the same features for all tasks, e.g., the same features are provided for gender classification and smile classification from facial images.

Furthermore, a producer cannot offer any arbitrary feature for any task. To claim a good performance over some offered tasks, it needs to train the whole pipeline in an end-to-end fashion to provide a meaningful feature for a certain task. The notations used throughout the paper to describe MetaMorphosis are shown in Table 2.

To construct the model in a cost-effective fashion and to reduce the model size as well, we divide the producer into a feature extractor part (encoder  $\mathcal{E}$ ), a MetaMorphosis module ( $\mathbf{g}(\cdot)$ ), and the target task. For clarity, we use the producer, feature extractor, and encoder as the same entity,  $\mathcal{E}$  or  $\mathcal{E}_p$  (encoder trained with differential privacy) throughout the paper. To produce meaningful features, the producer goes through a full training effort respecting the input obfuscation and task privacy. After training the whole model, the producer splits the model into two parts: one part includes a semi-universal encoder for some sub-tasks and unique transformer modules for each of the tasks. For other similar sub-tasks, another similar feature extractor module may exist. The remaining part will be hidden from the outside environment and is kept on the producer side only. Thus, the producer will offer access points for only subscribed consumers for the respective tasks. But where to split is an issue in maintaining communication vs. computation trade-off (see Sections 5 and 6). Although the earlier layers are suitable for a lightweight encoder, they may be prone to reconstruction image attack very easily. It is difficult to retain the original image from the layers closer to the outputs. The feature provider should also offer features so that consumers will produce the final output for a task with minimal effort.

**Table 2: Notation**

Description	Notation	Description	Notation
Input	$\mathbf{x}$	Output	$\mathbf{y}$
Producer Model	$\mathcal{M}_f$	Consumer Model	$\mathcal{M}_c$
Task	$T$	Task features	$\mathbf{z}_{1,2,\dots,T}$
Task Metamorphosis Module	$\mathbf{g}$	Encoder	$\mathcal{E}$
Private Encoder	$\mathcal{E}_p$	Private Feature	$\mathbf{z}_p$
MetaMorphosis	$\mathcal{G}$	Decoder	$\mathcal{E}^{-1}$

To design MetaMorphosis, suppose a model  $\mathcal{M}_f$  is trained by the producer that provides features for a corresponding task  $T_i$  for an input  $\mathbf{x}$ . At the time of inference, the producer will share the intermediate features as output, denoted by  $\mathbf{z}_i$ , from a portion of the model  $\mathcal{G}_i$  where  $\mathcal{G}_i \in \mathcal{M}_f$ . After getting the features  $\mathbf{z}_i$  instead of the raw input  $\mathbf{x}$ , the consumer runs its own model  $\mathcal{M}_c$  on  $\mathbf{z}_i$  and produces  $\hat{\mathbf{y}}_i$  for task  $T_i$  where the ground truth is  $\mathbf{y}_i$ . Mathematically, it can be written as follows.

$$\begin{aligned} \min \mathbf{y}_i \sim \hat{\mathbf{y}}_i &= \mathcal{M}_c \circ (\mathcal{G}_i \circ \mathbf{x}) = \mathcal{M}_c(\mathbf{z}_i) \\ \text{s.t. } \text{Acc}(T_i|\mathbf{z}_i) - \sum_{i \neq j} \text{Acc}(T_j|\mathbf{z}_i) &\approx \text{Acc}(T_i|\mathbf{z}_i) \quad (1) \\ \text{Acc}(T_i|\mathbf{z}_i) &\geq \xi \quad ; \quad 0 < \xi < 1 \end{aligned}$$

In Equation 1, the objective is to maintain the target performance ( $\text{Acc}$ ) for task  $T_i$  and obfuscate the input ( $\mathbf{x}$ ) and features ( $\mathbf{z}$ ) to limit the accuracy of all other tasks with the current task features. As  $\mathcal{M}_c$  and  $\mathcal{G}_i$  will not be processed by the same party, a few privacy considerations should be established. Based on this, we can divide the overall producer construction into two components: (1) **Input obfuscation**, and (2) **Task-privacy**.

In the next subsections, we will investigate thoroughly Equation 1 in terms of input obfuscation and task privacy and discuss the final equation as shown in Equation 8.

### 3.2 Input Obfuscation

By input obfuscation, we mean the input should be made private so that the features provided by the producer cannot be converted back to the original input by an attacker. If  $\mathbf{z} = \mathcal{E}(\mathbf{x}; \theta_{\mathcal{E}})$ , then it is nearly impossible to find a  $\mathcal{E}^{-1}$  so that  $\mathcal{E}^{-1}(\mathbf{z}) = \mathbf{x}$ . To ensure input obfuscation, we propose an efficient use of differential privacy which is defined as follows.

*Definition 3.1.* If  $d$  and  $d'$  are two adjacent inputs of  $D$  that differ by at least one sample and they follow a certain condition such that

$$\Pr[\mathbf{f}(d) \in D] \leq e^\epsilon \Pr[\mathbf{f}(d') \in D] + \delta \quad (2)$$

where,  $\mathbf{f}$  is a randomized function, i.e.,  $\mathbf{f} : D \rightarrow \mathcal{R}$ , then  $\mathbf{f}$  satisfies  $(\epsilon, \delta)$  differential privacy (DP) [1].

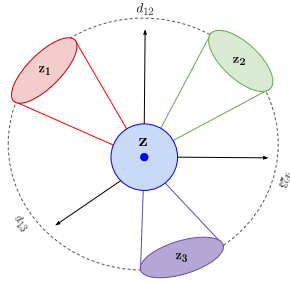
Definition 3.1 is also known as Rényi-differential privacy [28] which is a relaxed version of  $\epsilon$ -DP with a  $\delta$ . From Equation 2, we see that the higher the value of  $\epsilon$ , the lower the privacy bound. Differential privacy operations in deep learning models are shown in Algorithm 1 where noises are added with gradients before updating the parameters [1]. To get a desired  $\epsilon$ , the noise  $\sigma$  can be chosen for a number of training steps  $T$ , batch size  $q$ , and  $\epsilon < c_1 q^2 T$  as the following Equation 3 [1]. Here,  $c_1$  and  $c_2$  are constants.

$$\sigma \geq c_2 \frac{q \sqrt{T \log \frac{1}{\delta}}}{\epsilon} \quad (3)$$

Rather than adding differential privacy in the input as shown in recent literature [18], we perform DP in model parameters for input obfuscation. As in MetaMorphosis, the producer holds the feature extractor part only, to generate meaningful features, the feature extractors along with target classifiers are required to train jointly. A split learning method can reflect the scenario where a model is split into the feature extractor part and the classifier part. So, we

**Algorithm 1** Differential Privacy Operations

- 
- 1: **function** GRADIENT COMPUTATION(.)
  - 2:      $g_t(z_i) \leftarrow \nabla_{\theta_t} \mathcal{L}(\theta_t, z_i)$
  - 3: **end function**
  - 4: **function** GRADIENT CLIPPING(.)
  - 5:      $\bar{g}_t(z_i) \leftarrow g_t(z_i) / \max(1, \frac{\|g_t(z_i)\|_2}{C})$    ▶ Gradient Clipping  
with certain threshold  $C$
  - 6: **end function**
  - 7: **function** NOISE ADDITION(.)
  - 8:      $\hat{g}_t(z_i) \leftarrow \frac{1}{n} \sum_i (\bar{g}_t(z_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$    ▶ Adding noise to the  
gradient
  - 9: **end function**
- 

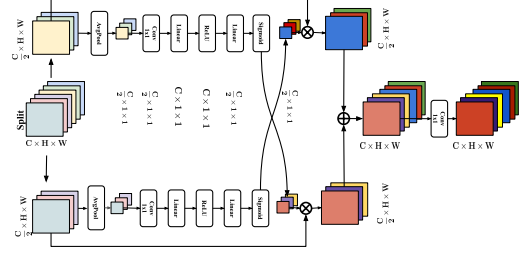
**Figure 2: Pictorial representation of privacy-aware task feature generation.**

propose differential privacy with split learning to achieve input obfuscation.

**Split learning with differential privacy:** Input obfuscation results in a trade-off between utility vs. privacy. As the main goal of MetaMorphosis is to provide task-specific features, injecting noises to ensure DP into the whole model parameters while training to ensure only input obfuscation to the encoder  $\mathcal{E}_p$  in unnecessary and it affects the task performance. In most cases, the consumer resides in the public domain and making the consumer model parameters private have little effect on overall privacy constraints. As a result, making only producer model parameters private will suffice our goal. In that case, during training of the feature generator, the provider uses differential privacy to ensure the input obfuscation of the generator only while learning the intended task using the split learning method [35]. A detailed discussion of utility vs. privacy is discussed in Section 4.3.

### 3.3 Task Privacy

As we consider the service provider (feature-provider) as an MLaaS (Machine Learning as a Service) platform, the service provider/producer will offer meaningful features for certain tasks to the public domain. In this case, instead of providing a single universal interface, the service provider offers multiple access points for some task-privacy-related features to the subscribers/consumers. By task privacy, we mean the features used for one task will not perform well for another task. Mathematically, the deep features generated for one task should be far apart from another task. We formally formulate

**Figure 3: Cross-SEC Metamorphosis Module**

task privacy as follows. For any input  $x$ , if there exist  $n$  feature extractors ( $\mathcal{G}_{1...n}$ ) for multi-task learning, then the optimal similarity between any two feature extractors  $\mathcal{G}_i(\cdot)$  and  $\mathcal{G}_j(\cdot)$  satisfies the following equation.

$$\sum_{i \neq j}^n \mathcal{S}(\mathcal{G}_i(x), \mathcal{G}_j(x)) \approx 0 \quad (4)$$

where  $\mathcal{S}(\cdot, \cdot)$  denotes a similarity function between two features. To ensure *task-privacy*, the summation of the similarity between features will be theoretically 0. A pictorial representation of task-privacy transformation on certain feature  $z$  is shown in Figure 2.

**Task Metamorphosis Module:** In MetaMorphosis, we propose a novel feature module for each task, instead of sharing a common feature for all consumers. The main goal of each metamorphosis module is to produce task-specific features as distinctively as possible with an assurance of better performance for the respective task. In this way, the attacker is unable to produce meaningful features for the specific private task. To ensure better performance, the metamorphosis module should capture the most informative feature of the task. To achieve this, we propose an attention-based metamorphosis module, Cross-SEC, that enables the general features  $\mathcal{E}$  to be more informative. At first, we split the  $\mathcal{E}(x)$  into  $k$  splits. For each split, we get the global attention of the features. Using the Conv-Linear-ReLU-Linear module, we transform the features and add a Sigmoid activation layer to get the attention values. Then, the attention is swapped between the splits following a Conv ( $1 \times 1$ ) layer. At the time of joint training of  $\mathcal{E}$  and  $g(\cdot)$ , the swapping of attention values will try to make  $\mathcal{E}$  more informative as it avoids making only some channels of  $\mathcal{E}$  more informative.

The metamorphosis module is shown in Figure 3 with the shape for each layer. To make the task features distinct, we use a similarity metric as used in recent literature [2, 18]. In this case, we use the SSIM metric to compare the structural similarity among task features, and in the loss function, it learns to project them far from each other based on the weight given to this metric. The task-privacy loss function can be written as follows.

$$\ell_{tp} = \sum_{i, j \in \mathcal{T}} \mathbb{1}[T_i \neq T_j] \mathcal{S}(g_i(\mathcal{E}(x)), g_j(\mathcal{E}(x))) \quad (5)$$

This metric will be added to the loss function with other task performance losses to achieve the desired behavior of MetaMorphosis.

Together, we can write the whole equation as follows.

$$\text{loss} = \sum_{i=1}^{|\mathbf{T}|} \mathcal{L}_i + \omega \sum_{i,j \in \mathbf{T}} \mathbb{1}[\mathbf{T}_i \neq \mathbf{T}_j] \text{SSIM}(\mathbf{g}_i(\mathcal{E}(\mathbf{x})), \mathbf{g}_j(\mathcal{E}(\mathbf{x}))) \quad (6)$$

Here,  $\omega$  controls the weight of the distance loss function to overall loss. To make the feature generator more efficient, we can use a single encoder and multi-task transformer modules for a group of tasks. To assure task privacy and input obfuscation, we can rewrite the function  $\mathcal{G}$  as a composition of private-encoder  $\mathcal{E}_p$  that prevents exposing the private information and a task transformer module that converts the  $\mathbf{g}$ . For task privacy only, the encoder can be non-private ( $\mathcal{E}$ ).

$$\mathcal{G}(\mathbf{x}) = (\mathbf{g} \circ \mathcal{E}_p)(\mathbf{x}) \quad (7)$$

We can combine these two aspects of privacy and elaborate the Equation 1 and relax the constraints to achieve efficient training as follows.

$$\begin{aligned} \min y_i \sim \hat{y}_i &= (\mathcal{M}_c \circ \mathcal{G}_i)(\mathbf{x}) = (\mathcal{M}_c \circ \mathbf{g}_i \circ \mathcal{E}_p)(\mathbf{x}) = \mathcal{M}_c(\mathbf{z}_i) \\ \text{s.t. } \mathcal{E}_p^{-1}(\mathbf{z}_i) &\neq \mathbf{x} \\ &\sum_{i \neq j}^{\mathbf{T}} \mathcal{S}(\mathcal{G}_i(\mathbf{x}), \mathcal{G}_j(\mathbf{x})) \approx 0 \end{aligned} \quad (8)$$

### 3.4 MetaMorphosis Training Scheme

The training scheme of MetaMorphosis is shown in Algorithm 2. MetaMorphosis obfuscates the input and the tasks in two phases. If input obfuscation and private attribute obfuscation are imposed, then the encoder with the privacy attribute classifier is trained jointly at Phase 1 [line 9 in Algorithm 2]. After the completion of Phase 1, in Phase 2, the task variant metamorphosis modules are trained along with the respective classifiers [line 10 in Algorithm 2], where the encoder trained from phase 1 is kept fixed to provide features. In line 9,  $\mathcal{M}_p$  refers to the private classifier (i.e. gender for face images) that the publisher intends to hide. It will train other tasks i.e.  $\mathcal{M}_{c_i}$  by hiding private information using task privacy [line 6-7, 10 in Algorithm 2]. After the completion of two-phase training, the task features are ready for the consumers. Figure 4 shows the steps of the training and inference scheme of MetaMorphosis. At the time of inference, the producer will offer access points ( $\mathbf{z}_{1,2,\dots,\mathbf{T}}$ ) for different tasks.

---

#### Algorithm 2 MetaMorphosis

---

- 1: **if** Input obfuscation only **then**
  - 2:    $\hat{y}_{c_i} = \mathcal{M}_{c_i} \circ \mathcal{E}_p \circ \mathbf{g}_i(\mathbf{x})$                      $\triangleright$  forward pass
  - 3:   Compute  $\mathcal{L}(\hat{y}_{c_i}, \hat{y}_{c_i})$
  - 4:   Update  $\theta_{\mathbf{g}_i}, \theta_{\mathcal{E}_p}$  using Algorithm 1, Update  $\theta_{c_i}$     $\triangleright$  backward pass
  - 5: **else if** Task-privacy only **then**
  - 6:   Compute  $\sum_{i=1}^{|\mathbf{T}|} \mathcal{L}_i + \omega \sum_{i \neq j} \text{SSIM}(\mathbf{g}_i(\mathbf{z}), \mathbf{g}_j(\mathbf{z}))$
  - 7:   Update  $\theta_{\mathbf{g}_i}, \theta_{\mathcal{E}}, \theta_{c_i} \forall i \in \mathbf{T}$
  - 8: **else**
  - 9:   At phase 1, do steps 2-4 to joint train the  $\mathcal{E}_p$  and  $\mathcal{M}_p$
  - 10:   At phase 2, using 6-7 train  $\mathbf{g}_i, \mathcal{E}_p$ , and  $\mathcal{M}_{c_i}$
  - 11: **end if**
- 

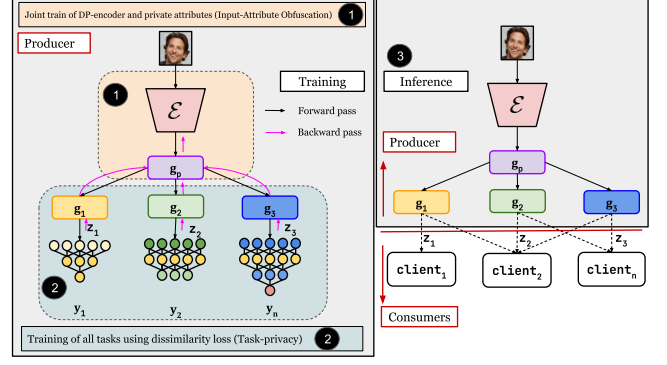


Figure 4: Producer training and inference scheme

### 3.5 Threat Model

Before going into detail on experimental results, we describe the attacker model in this section. For input obfuscation, we extract the private encoder features and use a decoder model (Figure 8) to act as an attacker trying to reconstruct the image. For task privacy, we assume the consumer portion of the model architecture is as same as the producer model architecture while training. In this, for all cases of classifiers, we use the same model architecture (ResNet-18) to act as an attacker. At the time of task privacy evaluation, we interchange the task metamorphosis module but keep the classifier layers and weights intact as the producer for the attacker. In Section 4, we implement and evaluate MetaMorphosis on different types of tasks and compare MetaMorphosis with recent relevant literature.

## 4 EVALUATION

### 4.1 Datasets and Metrics

To have a deep understanding of MetaMorphosis performance, we evaluate the task-privacy algorithm in different domains with various task complexities. To simulate an indoor robot scenario, we use the NYU-v2 dataset [29], which contains 1449 indoor images with ground truth images on three tasks, i.e., semantic segmentation, depth estimation, and surface normal estimation. We have used 795 images for training MetaMorphosis and evaluated the rest 654 images. To simulate the road scene-based tasks, we use the CityScapes dataset [9], which contains 3475 vehicle road scene views. Based on recent literature, we use the 2975 images for training and the rest 500 images for testing the performance of MetaMorphosis. We also use a large facial attribute dataset named CelebA [24] that includes more than 200000 images (162000 for training, 40000 for testing) to show multi-binary classification-based task privacy. For the multi-class classification scenario, we use StateFarm (a total of 22424 images, use 17934 for training and 4490 for testing) to validate the input obfuscation and task privacy.

### 4.2 Implementation details

We use PyTorch to implement MetaMorphosis and to execute the training we use 4 × 16 GB NVIDIA RTX A4000 workstation for all datasets. We use cross-entropy loss as shown below for segmentation and compute the mean Intersection over Union (mIoU), and

pixel accuracy as a performance metric as referred to [19].

$$\mathcal{L}_{seg} = -\frac{1}{m} \sum_{i=1}^m y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \quad (9)$$

For depth estimation, we use the absolute error as described by [19] and also use it in the loss function to minimize the depth error.

$$MAE_{depth} = \frac{1}{N} \frac{\sum_{i,j} |y_{i,j} - \hat{y}_{i,j}|}{\sum_{i,j} \mathbb{1}[y_{i,j} > 0]} \quad (10)$$

In surface normal estimation, we use the mean and median of per pixel error and compute the fraction of error within a certain threshold (11.25, 22.5, 30). For Cityscapes and NYU-v2, we use Adam optimizer with an initial learning rate (LR)  $1 \times 10^{-4}$ . A step learning rate scheduler changes the LR with step size 100 and  $\gamma = 0.5$ . We ran each experiment for 200 epochs and chose the best model with the smallest average training error of all tasks. We then use the best model for prediction. For NYU-v2, we use 13 classes for segmentation, and for Cityscapes, we use seven classes. The batch size is 8 and 2 for Cityscapes and NYU-v2, respectively. We use 0.001 as weight on SSIM loss while adding task privacy. For the CelebA and StateFarm datasets, we use  $\epsilon = 4$ , and 1.2 as the maximum gradient clipping ( $C$ ) for both StateFarm, and CelebA, and  $\delta = 10^{-5}$ ,  $10^{-6}$  respectively. The batch size is set to 64, and AdamW [26] optimizer with LR= $10^{-4}$ . For data transformation, we resize to make the images to  $64 \times 64$  pixels, and use RandomHorizontalFlip at training. The normalization parameters are used as same as ImageNet. We use Opacus [45] to train the model using differential privacy. We have chosen the lightweight ResNet-18 model, split it in half at different points, and used the first portion as the encoder and the rest as the private and intended classifier.

### 4.3 Experimental Results

**CityScapes and NYU-v2:** As MetaMorphosis imposes privacy constraints either on the content or on the task or on both. Considering task privacy we focus on NYU-v2 and Cityscapes dataset. For both datasets, we use SegNet model [19] with knowledge distillation during training. Table 3 shows the results on the test set using KD-MTL [19] where privacy-aware feature generation is absent. With the addition of cross-SEC metamorphosis module and SSIM loss function, we compare the utility as the performance metric for both and compare task privacy based on the interchange of the module. For having the distilled knowledge, we first train every single model to train a single task. Then using Algorithm 2 for task-privacy only, we train the joint model to produce output similar to every single model and add the privacy loss to make each task features distinct. We joint train the segmentation, depth, and surface-normal estimation for NYU-v2 using task-transformer module and compare it without the task-transformer module and without task privacy. For segmentation results, we observe a 7.61% higher mean Intersection Over Union (mIoU) than KD-MTL and a 2.25% higher pixel accuracy metric. Compared to the depth estimation results, MetaMorphosis achieves almost the same results for absolute error and a little worse in relative error. having a  $conv(1 \times 1)$  for each task. As cross-SEC transformer generalizes better task features. For State-Farm dataset, we train for 20 epochs. For CelebA we train for 10 epochs.

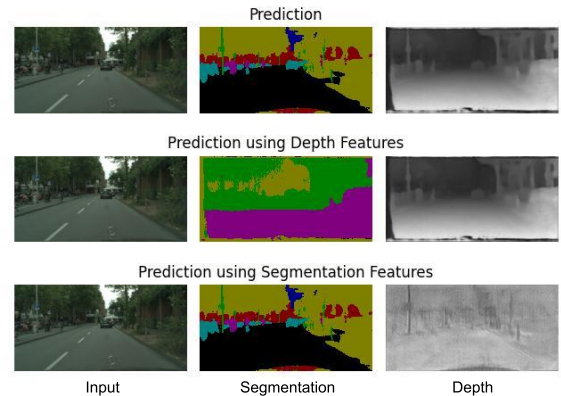
**Table 3: Test set results on CityScapes [9] dataset. MetaMorphosis achieves higher pixel accuracy for segmentation and almost the same absolute error for depth.**

Model	Size (MB)	Segmentation		Depth	
		mIoU ( $\uparrow$ )	Pix Acc ( $\uparrow$ )	Abs Err ( $\downarrow$ )	Rel Err ( $\downarrow$ )
KD-MTL [19]	300.90	52.18	91.24	0.0140	28.90
MetaMorphosis	307.00	<b>59.79</b>	<b>93.49</b>	0.0141	31.89

**Table 4: Task-privacy evaluation of Cityscapes [9]. Use of one task metamorphosis module to evaluate the performance of other tasks. Using depth features for segmentation, lower mIoU, and pixel accuracy indicate higher task privacy and vice versa. For depth estimation, the higher error with segmentation features indicates higher task privacy and vice versa.**

Metamorphosis Module (Replaced)	Methods	Segmentation		Depth	
		mIoU ( $\uparrow$ )	Pix Acc ( $\uparrow$ )	Abs Err ( $\downarrow$ )	Rel Err ( $\downarrow$ )
—	MetaMorphosis	59.79	93.49	0.0141	31.89
Segmentation	MetaMorphosis	59.79	93.49	<b>0.1079</b>	<b>99.07</b>
Depth	MetaMorphosis	<b>1.47</b>	<b>7.33</b>	0.0141	31.89

To show the task-privacy evaluation, we use the trained segmentation Cross-SEC module features to infer segmentation and depth estimation and vice versa for depth estimation. From Table 4, we observe a sharp drop in the performance of both tasks. For segmentation, the mIoU and pixel accuracy drop down to 1.47% and 7.33%, respectively. For depth estimation, the absolute error is almost 10 $\times$  higher using the segmentation feature.



**Figure 5: Qualitative analysis of task privacy. Prediction results of the segmentation and depth estimations of the CityScapes dataset using MetaMorphosis (first row). The task metamorphosis modules are interchanged and the output is produced (second and third row).**

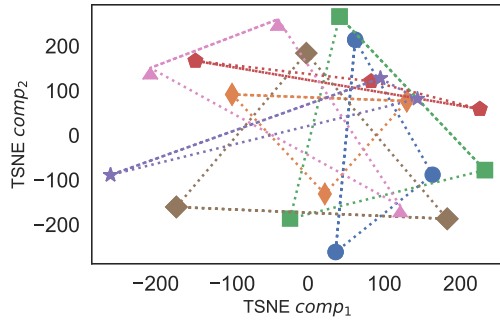
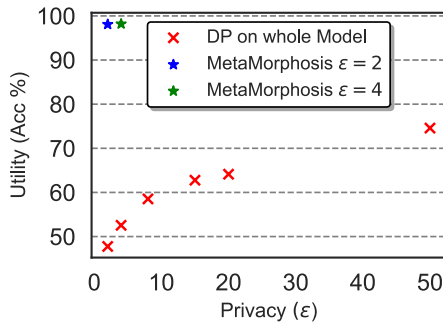
We also observe the qualitative results of CityScapes using task privacy as shown in Figure 5. The segmentation and depth estimation outputs are almost obscured if respective features are not used for respective tasks. To evaluate more complicated tasks, we

**Table 5: Test results on NYU-v2 dataset. In spite of imposing task privacy, MetaMorphosis achieves almost the same performance as [19]. STL refers to the results of single-task learning models.**

Model	Size (MB)	Methods	Segmentation		Depth		Surface Normal				
			mIoU ( $\uparrow$ )	Pix Acc ( $\uparrow$ )	Abs Err ( $\downarrow$ )	Rel Err ( $\downarrow$ )	Mean ( $\downarrow$ )	Median ( $\downarrow$ )	11.25 ( $\uparrow$ )	22.5 ( $\uparrow$ )	30 ( $\uparrow$ )
SegNet	300.90	STL	17.32	55.70	0.6577	0.2828	29.99	23.81	24.31	48.06	60.05
	300.90	KD-MTL [19]	18.75	58.02	0.5780	0.2467	29.40	23.71	24.33	48.22	60.45
	310.8	MetaMorphosis	18.14	57.03	0.5867	0.2498	30.47	24.73	22.92	46.50	58.62

**Table 6: NYU-v2 [29] task-privacy evaluation by interchanging metamorphosis module.**

MetaMorphosis Module	Methods	Segmentation		Depth		Surface Normal				
		mIoU ( $\uparrow$ )	Pix Acc ( $\uparrow$ )	Abs Err ( $\downarrow$ )	Rel Err ( $\downarrow$ )	Mean ( $\downarrow$ )	Median ( $\downarrow$ )	11.25 ( $\uparrow$ )	22.5 ( $\uparrow$ )	30 ( $\uparrow$ )
—	MetaMorphosis	18.14	57.03	0.5867	0.2498	30.47	24.73	22.92	46.50	58.62
Segmentation	MetaMorphosis	18.14	57.03	<u>1.2541</u>	<u>0.5014</u>	<u>51.97</u>	<u>51.37</u>	<u>1.74</u>	<u>9.14</u>	<u>17.89</u>
Depth		<u>4.27</u>	<u>18.28</u>	0.5867	0.2498	<u>54.77</u>	<u>54.38</u>	<u>4.41</u>	<u>14.41</u>	<u>22.02</u>
Surface-normal		<u>3.37</u>	<u>16.56</u>	<u>1.8694</u>	<u>0.7843</u>	30.47	24.73	22.92	46.50	58.62

**Figure 6: Task-oriented feature projection using t-SNE on NYU-v2 dataset. The triangle points having the same color refer to segmentation, depth, and surface normal features for the same input. The distant feature position for the same input verifies task privacy in t-SNE.****Figure 7: MetaMorphosis achieves better utility whereas respecting input obfuscation. Higher privacy ensures if  $\epsilon \rightarrow 0$ . So, the best points having high utility and high privacy locates in the top left quadrant.**

consider adding surface normal estimation with the segmentation and depth tasks and impose task privacy. We use NYU-v2 dataset in this regard. We have found similar results as on Cityscapes dataset. In NYU-v2, we also observe similar performance as compared to KD-MTL [19] with a little deflection in performance metric ( $\sim < 1\%$  for segmentation and depth estimation, and  $\sim < 2\%$  for surface normal estimation as shown in Table 5. We also evaluate task privacy on NYU-v2 by interchanging the metamorphosis modules as shown in Table 6. The mIoU for segmentation drops down to  $3.37 \sim 4.27$ , the absolute depth error rises to  $>1$ , and the mean value of surface normal goes high from 30.47 to  $51.97 \sim 54.77$ . To evaluate the task-specific feature projection, we investigate the inference of the model trained on NYU-v2 and project the three task features using t-SNE representation as shown in Figure 6. The task features for each input are projected by training them using t-SNE. We show the task projection points in the same color and form a triangle to observe how separate they are. The higher area of the triangle means a higher distance. The component values of t-SNE additionally illustrate the distance among feature projections for each input.

**StateFarm:** To evaluate MetaMorphosis in achieving task utility and input obfuscation, we use the distracted driver recognition task having 10 classes. At first, we impose differential privacy (DP) into the model encoder and classifier part. By varying the  $\epsilon$ , we compute the distracted behavior recognition accuracy. In Figure 7, we observe the accuracy drops with the increase in privacy (In DP, the  $\epsilon \rightarrow 0$  ensured higher privacy and vice-versa). According to Algorithm 2, to ensure input obfuscation, by adding DP-guarantee to the encoder side only, MetaMorphosis achieves both utility and privacy. We also evaluate differential privacy qualitatively. In Figure 8, reconstruction of the Statefarm dataset is performed using an encoder and decoder. Using the encoder features, the distracted driver recognition task is performed. The decoder succeeds in decoding the image. Then, we train the encoder using differential privacy. In this case, the decoder fails to reconstruct images. Even using the private encoder, we train a decoder to reconstruct the image.



**Table 7: Test results on input obfuscation and task-privacy on Statefarm dataset**

Task Metamorphosis Module	Classifier	DeepObfuscator [18]	MetaMorphosis
Identity	Identity	99.97	100.00
Behavior	Identity	30.28	<b>1.49</b>
Behavior	Behavior	98.32	98.69
Identity	Behavior	—	<b>10.34</b>

Even after training, the decoder failed, but we got 98.69% accuracy for the intended distracted driver recognition task. In Figure 8, we observe the DP-guarantee of obfuscating deep features in spite of training a decoder with the obfuscated features.

In addition to a private attribute, we first train the encoder and the private attribute task jointly using DP on the encoder. After the models are trained, then we train the intended classification task using MetaMorphosis, where a task-transformer module is trained using DP, and it generates distinct features for the distracted driver recognition task from private features by adding MS-SSIM to the loss function. In this way, the full process will maintain content-task privacy. Table 7 shows the evaluation of the final output of driver identity recognition and distracted driver behavior recognition tasks. After training with MetaMorphosis, for driver identity recognition, we got 100%, and for distracted driver behavior detection, we got 98.69% accuracy. Then by interchanging the task-transformer module, we compute the accuracy of each task which implies task privacy. We observe only 1.49% accuracy if an intruder use behavior features for driver identity classification. In comparison to deepObfuscator [18], considering behavior features as general features to shared (as private features will be hidden in feature producer), MetaMorphosis achieves only 1.49% accuracy for driver identity recognition whereas, for deepObfuscator, it achieves 30.38% accuracy. So, MetaMorphosis ensures more privacy in obfuscating private attributes.

**Table 8: Test results on task privacy and input obfuscation on CelebA**

split point	Task Metamorphosis Module	Classifier	DeepObfuscator	MetaMorphosis
5	Gender	Gender	—	95.94
	Smile	Gender	55.85	<b>34.56</b>
	Smile	Smile	89.52	89.89
	Gender	Smile	—	<b>42.49</b>

**CelebA:** To validate the task privacy and input obfuscation jointly, We have experimented with CelebA dataset. We consider two scenarios (1) smile, gender, and (2) smile, gender, cheekbone classification where gender is a private attribute and input obfuscation is imposed. To achieve this, according to Algorithm 2, we joint train the encoder and the private gender classifier first. For this, we also use a comparatively smaller model, ResNet-18, and split it into different positions to build the encoder and the classifier. Without loss of generality, we use the same classifier model size for all tasks. After training of encoder with DP and the private classifier, we use the task-privacy loss to train the classifier for smile for case (1) and

the smile and cheekbone classifier jointly for case (2). Then, we test the performance of all tasks and task privacy by interchanging the task-metamorphosis modules. It is to be noted that gender features are created so that we can make other attributes' features distinct from the private features, and this private attribute feature will be hidden and kept on the producer side. As in MetaMorphosis, we show that one private attribute defined for one task may be defined as non-private by another task. In Table 8, we consider gender as private and the smile classification as the intended classification task. Both private and non-private classifiers achieve almost similar performance as DeepObfuscator [18] but hide privacy information better (21.29% less accuracy than DeepObfuscator). In this case, MetaMorphosis achieved 34.56% accuracy while doing gender classification using the smile classifier. The reason for a bit increase in gender accuracy with one task (smile) and two tasks (smile and cheekbone) indicates an intrinsic correlation between the two tasks as discussed by recent literature [18]. In this case, MetaMorphosis diminishes the correlation more than DeepObfuscator.

We investigate the second scenario where the number of intended tasks is two. In this, we adopt smile and cheekbone classification as two intended tasks. In this scenario, the privacy requirements are similar to the previous one i.e. input obfuscation and task privacy. MetaMorphosis classifies smile and cheekbone while obfuscating the gender attribute and input. In Figure 9, we achieve 89.78 ~ 89.89% accuracy for smile and 83.02 ~ 84.2% accuracy for cheekbone classification using a variety of weight  $\omega$  of task-privacy while ensuring input obfuscation using DP. MetaMorphosis has achieved similar results for intended task classification but hides privacy 8.69% ~ 20.42% more than DeepObfuscator [18] using smile classification task features and 11.69% ~ 20.12% using cheekbone classification task features. Relation between  $\omega$  and accuracy basically depends on task-correlation and is interesting to investigate which we keep as our future work.

## 5 SYSTEM DEPLOYMENT

For system deployment, the producer will send features, and the consumer will produce the outputs. But to build such a setup, the consumer needs to know the ground truth to compute the loss function. If the features and labels are shared for joint training, the attacker can eavesdrop on the features and design any system to capture the feature output relationship. Instead, the labels are sent at a specific time using an encryption key. Then, the features can be shared to do the training task. To replicate the scenario, we consider a Quadro RTX 4000 as the producer (server) and a Raspberry-pi as well as a Jetson Nano as the consumers (client) in Figure 10. At the forward pass of training, the producer sends the intermediate features to the consumers. The consumers produce the output, compute the loss function and send the computed gradient in the backward pass. Based on the gradient, the producer updates its parameters. Based on input obfuscation, the producer is trained with a DP-optimizer.

We demonstrate a split neural network using ResNet-18 model with different indices as split points and execute the training. As an MLaaS platform, the service provider should offer features such that the consumer can do the task with little effort. With the smaller

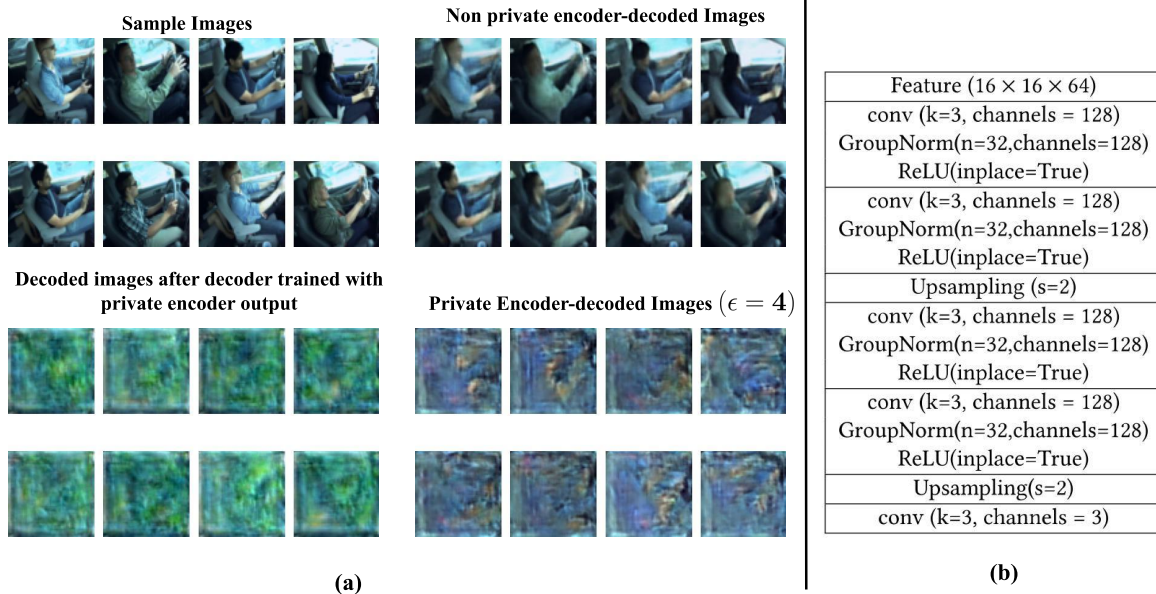


Figure 8: Input obfuscation on the StateFarm dataset. (a) Top left: Sample images. Top right: images decoded from a non-private encoder. Bottom right: images decoded from a private encoder trained with differential privacy having  $\epsilon = 4$ ,  $\delta = 1e^{-4}$  and a decoder (Figure 8b). Bottom left: Images generated by a decoder after training using the same private encoder. (b) Decoder model by the attacker for image reconstruction attack on private features.

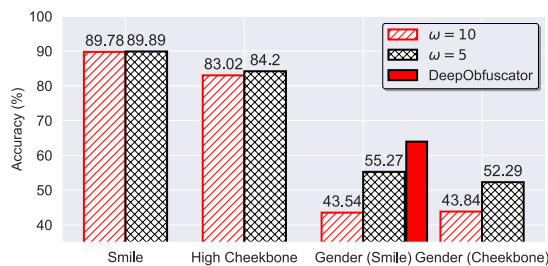


Figure 9: Smile and cheekbone classification from CelebA while obfuscating gender attribute using MetaMorphosis.

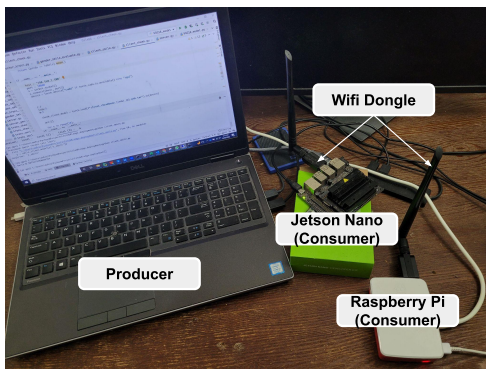


Figure 10: Producer-Consumer Deployment

Table 9: Round trip time of sending features and collecting the gradients vs the consumer (client) model size vs the intermediate feature size.

Server Model (MB)	Client Model (MB)	Raspberry Pi RTT (ms)	Jetson nano RTT (ms)	Feature	Feature size KB
42.66	0.02	3.95	0.73	$512 \times 1 \times 1$	1.99
10.64	32.04	112.67	10.11	$256 \times 4 \times 4$	14.86
0.61	42.07	173.23	18.60	$64 \times 16 \times 16$	65.92

consumer model size, the round trip time will be lower for the accumulation of gradients by the producer. Table 9 shows the higher the round trip time with the higher feature, the larger the client model size. Due to the usage of GPU by Nano, the difference between the round-trip time of Jetson Nano and Raspberry-Pi is significant.

We also investigate the effect of adding a task metamorphosis module to the overall server latency. The latency of the metamorphosis module depends on the input shared by the encoder and its size. Figure 10 shows the effect of inference latency of adding the metamorphosis module with the encoder for running the smile classification task with the encoder, smile metamorphosis module, and the classifier. The little difference in latency proves the metamorphosis module to be a lightweight one.

## 6 ABLATION STUDY

We evaluate the performance of the Cross-SEC module, and we also experiment without crossing the connections after getting the attention of one portion of the features. In Table 11, joint training of segmentation and depth estimation is done where the task-privacy

**Table 10: Effect of MetaMorphosis module to Server latency**

MetaMorphosis Module	Model	Split Index	Server Size	MetaMorphosis Module Size	Server (ms) Latency (ms)
✗ ✓	ResNet-18	5	640.60 KB	54.5 KB	0.068 0.106
✗ ✓	ResNet-18	7	11.20 MB	791.80 KB	0.244 0.265
✗ ✓	ResNet-18	8	44.70 MB	3.20 MB	0.293 0.359

module is the Cross-SEC module and SEC module where no cross-connection between features occurs. Cross-SEC morphosis module performs better in achieving the metrics for segmentation and depth estimation than the SEC module. Regarding task privacy, cross-SEC achieves lower pixel accuracy, mIOU for segmentation with depth features, and lower absolute error for depth estimation with segmentation features than SEC module.

**Table 11: Importance of Cross-SEC module over SEC module without cross attention**

MetaMorphosis Module	Methods	Segmentation		Depth	
		mIoU (↑)	Pix Acc (↑)	Abs Err (↓)	Rel Err (↓)
–	MTL-SEC	57.79	93.39	0.0148	45.07
Segmentation	MTL-SEC	57.79	93.39	0.1022	110.09
Depth		3.92	23.97	0.0148	45.07
–	MTL-Cross-SEC	<b>59.79</b>	<b>93.49</b>	<b>0.0141</b>	<b>31.89</b>
Segmentation	MTL-Cross-SEC	59.79	93.49	0.1079	99.07
Depth		1.47	7.33	0.0141	31.89

**Table 12: Input-attribute obfuscation trade-off**

Method	Model	Provider Size (MB)	Classifier Size (MB)	MetaMorphosis Module	Classifier	Accuracy (%)
DeepObfuscator [18]	VGG16	1.0	536	universal	gender	55.85
MetaMorphosis	ResNet18	3.00	42.00	gender	gender	95.44
				smile	smile	87.78
				gender	smile	42.69
				smile	gender	45.14
		11.97	33.60	gender	gender	94.19
				smile	smile	82.09
				gender	smile	42.56
				smile	gender	61.35
		47.9	0.006	gender	gender	92.74
				smile	smile	58.83
				gender	smile	42.33
				smile	gender	38.65

To observe the trade-off between input and privacy attribute obfuscation, we change the encoder and classifier size by changing the split index of the ResNet-18 model. We identify an increase in privacy attribute leakage with the decrease of the classifier model size and expansion of the encoder model size (Table 12). We have found the gender classifier accuracy 45.14%, and 61.25 with lowering the classifier size from 42 MB to 33.6 MB. It is even worse for a classifier having 0.006 MB. We observe that with the increase of the encoder model and the decrease of the classifier model, it is difficult for the intended classification task to meet the input obfuscation and task privacy together. As more noise is fed to the encoder model to maintain the  $\epsilon$ -DP while training, less performance is desired with the decreased classifier model size, as also evident from Figure 7. DeepObfuscator used VGG-16 where only 1 MB portion is defined

as feature provider, and the 536 MB is designated to the intended class classifier whereas in MetaMorphosis, even using a small model ResNet-18, with higher encoder size, we achieve almost the same accuracy as DeepObfuscator and hides privacy attribute better by lowering gender classification task. Finding the optimal split index between the encoder and the task classifier is an interesting area to achieve input obfuscation and task privacy. We have kept this discussion as our future work.

## 7 RELATED WORK

Various methods for solving complex segmentation and depth estimation-based multi-task learning are discussed in the literature [4, 31, 40]. A knowledge distillation technique is proposed by Liu et al. [23] specifically for semantic segmentation tasks. Nguyen et al. [30] proposed a convolutional neural network to identify modified images, and the trained network can give a segmented mask for the modified region. An empirical study has been conducted by Standley et al. [34] to identify the factors that influence the performance of multi-task learning and proposed a framework to limit the number of multi-task models based on the correlation of tasks. SSIM [41] provides an image quality assessment metric called structural similarity (SSIM) to evaluate the similarity between two images. It can also be used as a loss function to impose dissimilarity between features by shifting the value close to zero. Attention modules are proposed in the literature [39, 47] to capture important features for target accuracy without dimensionality reduction. Chen et al. [6] propose a gradient normalization algorithm for training multi-task models to balance the training processes of different tasks. The algorithm improves accuracy and decreases the over-fitting effects for various kinds of tasks and on different datasets. Transformer-based cross-task attention mechanism [25] projects the features of one task to another. But the notion of distinct feature generation to achieve privacy is absent. In collaborative intelligence, to build a more efficient system, many layer output compression methods [7, 8, 44] and gradient compression methods [36] are suggested. Other [38, 48] focuses on multi-task feature compression. These compression techniques, referred to as Video Coding for Machine (VCM) [43], aims to reduce the communication overhead while maintaining the system performance, while many efforts [14, 15, 22, 32, 46, 47] are also devoted to optimizing the computational overhead. On the other hand, to build a good collaborative intelligent system, besides improving its efficiency, privacy-preserving feature encoding schemes also need to protect the privacy of data holders. In the case of differential privacy in deep learning, the DP-SGD algorithm was proposed by Abadi et al. [1]. Many variants of differential privacy, such as label differential privacy, are discussed in [12]. Table 1 illustrates the comparison of MetaMorphosis with recent similar literature.

## 8 CONCLUSION

In this paper, we propose MetaMorphosis that enables input obfuscation and task privacy for multi-task learning in a collaborative intelligence setup. In this paper, the main focus lies in sharing data and computation securely between a deep feature provider-based MLaaS platform and a number of consumers who subscribe to the provider according to interest. To ensure this, MetaMorphosis has

gone through a two-phase training scheme where the first phase ensures input privacy and private attribute privacy. Then the second training phase ensures task privacy among shared tasks through a unique squeeze-excitation based MetaMorphosis module. Experimental results on different domain datasets show the supremacy of MetaMorphosis over recent multi-task and adversarial learning methods. The MetaMorphosis also positively affects the sequential addition of new tasks in a multi-task environment because of its two-phase training scheme. This paper opens up some questions and disadvantages of having a universal feature for a split learning system as well as in a split federated learning system. As the performance of a federated learning system mostly depends on the *honesty* of the clients, the intuitive creation of task features despite the task *correlation* is still challenging with the increase in the number of tasks. In the future, we will focus on how task-variant features can be used to enable more privacy in federated learning systems.

## ACKNOWLEDGMENTS

We thank our anonymous reviewers and our shepherd Mimi Xie for their valuable feedback.

## REFERENCES

- [1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 308–318.
- [2] Md Adnan Arefeen, Sumaiya Tabassum Nimi, Md Yusuf Sarwar Uddin, and Yuyung Lee. 2021. TransJury: Towards Explainable Transfer Learning through Selection of Layers from Deep Neural Networks. In *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, 978–984.
- [3] Md Adnan Arefeen, Sumaiya Tabassum Nimi, Md Yusuf Sarwar Uddin, and Zhu Li. 2021. A lightweight relu-based feature fusion for aerial scene classification. In *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE, 3857–3861.
- [4] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. 2021. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4009–4018.
- [5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*. 801–818.
- [6] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. 2018. GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*. PMLR, 794–803.
- [7] Hyomin Choi and Ivan V Bajić. 2018. Deep feature compression for collaborative object detection. In *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 3743–3747.
- [8] Robert A Cohen, Hyomin Choi, and Ivan V Bajić. 2020. Lightweight compression of neural network feature tensors for collaborative intelligence. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 1–6.
- [9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3213–3223.
- [10] Xiaofeng Ding, Hongbiao Fang, Zhilin Zhang, Kim-Kwang Raymond Choo, and Hai Jin. 2020. Privacy-preserving feature extraction via adversarial training. *IEEE Transactions on Knowledge and Data Engineering* (2020).
- [11] Mohammed AM Elhassan, Chenxi Huang, Chenhui Yang, and Tewodros Legesse Munea. 2021. DSANet: Dilated spatial attention for real-time semantic segmentation in urban street scenes. *Expert Systems with Applications* 183 (2021), 115090.
- [12] Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, and Chiyuan Zhang. 2021. Deep learning with label differential privacy. *Advances in Neural Information Processing Systems* 34 (2021), 27131–27145.
- [13] Google. 2022. Google video intelligence API. <https://cloud.google.com/video-intelligence/>.
- [14] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132–7141.
- [15] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. *arXiv preprint arXiv:1602.07360* (2016).
- [16] Apoorv Khattar, Srinidhi Hegde, and Ramya Hebbalaguppe. 2021. Cross-domain multi-task learning for object detection and saliency estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3639–3648.
- [17] Ang Li, Yixiao Duan, Huanrui Yang, Yiran Chen, and Jianlei Yang. 2020. TIPRDC: task-independent privacy-respecting data crowdsourcing framework for deep learning with anonymized intermediate representations. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 824–832.
- [18] Ang Li, Jiayi Guo, Huanrui Yang, Flora D Salim, and Yiran Chen. 2021. Deep-Obfuscator: Obfuscating intermediate representations with privacy-preserving adversarial learning on smartphones. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*. 28–39.
- [19] Wei-Hong Li and Hakan Bilen. 2020. Knowledge distillation for multi-task learning. In *European Conference on Computer Vision*. Springer, 163–176.
- [20] Wei-Hong Li, Xialei Liu, and Hakan Bilen. 2021. Universal representation learning from multiple domains for few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9526–9535.
- [21] Wei-Hong Li, Xialei Liu, and Hakan Bilen. 2022. Universal Representations: A Unified Look at Multiple Task and Domain Learning. *arXiv preprint arXiv:2204.02744* (2022).
- [22] Xiang Li, Wenhui Wang, Xiaolin Hu, and Jian Yang. 2019. Selective kernel networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 510–519.
- [23] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. 2019. Structured knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2604–2613.
- [24] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep Learning Face Attributes in the Wild. In *Proceedings of International Conference on Computer Vision (ICCV)*.
- [25] Ivan Lopes, Tuan-Hung Vu, and Raoul de Charette. 2022. Cross-task Attention Mechanism for Dense Multi-task Learning. *arXiv preprint arXiv:2206.08927* (2022).
- [26] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101* (2017).
- [27] Microsoft. 2022. Microsoft computer vision API. <http://azure.microsoft.com/en-us/products/cognitive-services/computer-vision/>.
- [28] Ilya Mironov. 2017. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*. IEEE, 263–275.
- [29] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. 2012. Indoor Segmentation and Support Inference from RGBD Images. In *ECCV*.
- [30] Huy H Nguyen, Fuming Fang, Junichi Yamagishi, and Isao Echizen. 2019. Multi-task learning for detecting and segmenting manipulated facial images and videos. In *2019 IEEE 10th International Conference on Biometrics Theory, Applications and Systems (BTAS)*. IEEE, 1–8.
- [31] Sumaiya Tabassum Nimi, Md Adnan Arefeen, Md Yusuf Sarwar Uddin, Biplob Debnath, and Sriram Chakradhar. 2022. Chimera: Context-aware splittable deep multitasking models for edge intelligence. In *2022 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 70–77.
- [32] Xuran Pan, Chunjiang Ge, Rui Lu, Shiji Song, Guanfu Chen, Zeyi Huang, and Gao Huang. 2022. On the integration of self-attention and convolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 815–825.
- [33] Daniel Seichter, Mona Köhler, Benjamin Lewandowski, Tim Wengefeld, and Horst-Michael Gross. 2021. Efficient rgb-d semantic segmentation for indoor scene analysis. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 13525–13531.
- [34] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. 2020. Which tasks should be learned together in multi-task learning?. In *International Conference on Machine Learning*. PMLR, 9120–9132.
- [35] Chandra Thapa, Pathum Chamikara Mahawaga Arachchige, Seyit Camtepe, and Lichao Sun. 2022. Splitfed: When federated learning meets split learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 8485–8493.
- [36] Thijs Vogels, Sai Praneeth Karimireddy, and Martin Jaggi. 2019. PowerSGD: Practical low-rank gradient compression for distributed optimization. *Advances in Neural Information Processing Systems* 32 (2019).
- [37] Li Wang, Dong Li, Han Liu, Jinzhang Peng, Lu Tian, and Yi Shan. 2022. Cross-dataset collaborative learning for semantic segmentation in autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 2487–2494.
- [38] Mengyang Wang, Zhicong Zhang, Jiahui Li, Mengyao Ma, and Xiaopeng Fan. 2021. Deep joint source-channel coding for multi-task network. *IEEE Signal Processing Letters* 28 (2021), 1973–1977.
- [39] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. 2020. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- [40] Yu Wang, Quan Zhou, Jia Liu, Jian Xiong, Guangwei Gao, Xiaofu Wu, and Longjin Jan Latecki. 2019. Lednet: A lightweight encoder-decoder network for real-time semantic segmentation. In *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 1860–1864.
- [41] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.
- [42] Taihong Xiao, Yi-Hsuan Tsai, Kihyuk Sohn, Manmohan Chandraker, and Ming-Hsuan Yang. 2020. Adversarial learning of privacy-preserving and task-oriented representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 12434–12441.
- [43] Wenhan Yang, Haofeng Huang, Yueyu Hu, Ling-Yu Duan, and Jiaying Liu. 2021. Video Coding for Machine: Compact Visual Representation Compression for Intelligent Collaborative Analytics. *arXiv preprint arXiv:2110.09241* (2021).
- [44] Shuochoao Yao, Jinyang Li, Dongxin Liu, Tianshi Wang, Shengzhong Liu, Huajie Shao, and Tarek Abdelzaher. 2020. Deep compressive offloading: Speeding up neural network inference by trading edge computation for network latency. In *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*. 476–488.
- [45] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. 2021. Opacus: User-friendly differential privacy library in PyTorch. *arXiv preprint arXiv:2109.12298* (2021).
- [46] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. 2022. Resnest: Split-attention networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2736–2746.
- [47] Qing-Long Zhang and Yu-Bin Yang. 2021. Sa-net: Shuffle attention for deep convolutional neural networks. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2235–2239.
- [48] Zhicong Zhang, Mengyang Wang, Mengyao Ma, Jiahui Li, and Xiaopeng Fan. 2021. Msfc: Deep feature compression in multi-task network. In *2021 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 1–6.