Li-ion Battery Material phase prediction through Hierarchical Curriculum Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Li-ion Batteries (LIB), one of the most efficient energy storage devices, are widely adopted in many industrial applications. Imaging data of these battery electrodes obtained from X-ray tomography can explain the distribution of material constituents and allow reconstructions to study electron transport pathways. Therefore, it can eventually help quantify various associated properties of electrodes (e.g., volume-specific surface area, porosity) which determine the performance of batteries. However, these images often suffer from low image contrast between multiple material constituents , making it difficult for humans to distinguish and characterize these constituents through visualization. A minor error in detecting distributions among the material constituents can lead to a high error in the calculated parameters of material properties. We present a novel hierarchical curriculum learning framework to address the complex task of estimating material constituent distribution in battery electrodes. To provide spatially smooth prediction, our framework comprises three modules: (i) an uncertainty-aware model trained to yield inferences conditioned upon global knowledge of material distribution, (ii) a technique to capture relatively more fine-grained (local) distributional signals, (iii) an aggregator to appropriately fuse the local and global effects towards obtaining the final distribution. Code and data are available online.[1]

1 Introduction

Lithium Ion batteries (LIB) are an advanced energy storage technology employed across many highly impactful mechanical and technological applications ranging from electric vehicles to smartphones and laptops. A critical facet of the functioning of LIB is the two electrodes (i.e., anode and cathode) that enable the storage and flow of electricity. These electrodes are typically composite materials consisting of electro-chemically active material particles (e.g., Li, Ni, Mn) and a polymeric binder (e.g., C) to bind the active material particles onto a substrate (typically a thin aluminum sheet). During battery manufacture, these ingredients (i.e., active material particles and binder) are mixed in solvent and finally *coated* onto the thin aluminum substrate. The coating and the substrate together form the electrode.

Motivation: The microstructure of these composite electrode coatings significantly influences the LIB performance. Specifically, homogeneity of the coating *thickness* across the entire electrode ensures optimal electrical conductivity. Since the primary purpose of the *binder* used in electrodes is to hold the active material in the electrodes, the distribution of the binder throughout the electrode coating determines the thickness of the coating. The lack of homogeneous distribution of the binder

[1] Code Repository: https://bit.ly/3HJ2xZf

Submitted to NeurIPS 2021 AI for Science Workshop.

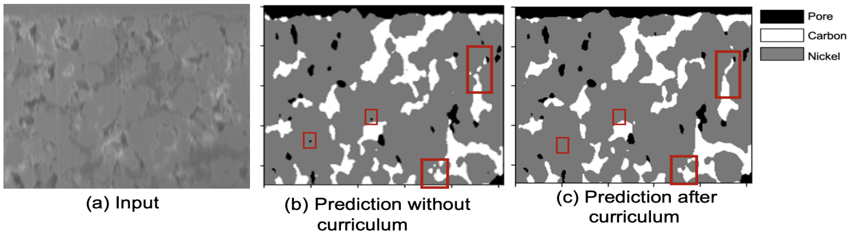
Pore
Carbon
Nickel
(a) Input
(b) Prediction without curriculum
(c) Prediction after curriculum

Figure 1: Motivation for MatPhase. (a) shows a micro-scale image of a portion of electrode sample. The varying intensities across the image represents possible transitions among material constituents which cannot be easily perceived by human. (b) Heatmap of predictions for the corresponding pixels obtained through a popular image segmentation model Ronneberger et al. [2015] (grey, white and black pixels denote Ni, C, and pore respectively). (c) Shows the output after adopting our proposed hierarchical curriculum learning. The red rectangles indicate some of the prediction differences between (b) and (c). Predictions at (c) are more smooth and free of artifacts.

(and thereby the active material) can also lead to the absence of electrode coating in certain regions termed *pores*, thereby impeding the current flow in those regions, causing degradation in battery performance.

Many approaches have been proposed to account for the binder distribution but are all heuristic in nature. Despite the advancements in imaging (e.g., X-ray microtomography), there has been noticeable difficulty in resolving the location of all phases (i.e., active materials, binder, pores) within the microstructure Pietsch et al. [2018]. Since, existing imaging tools (at micro-scale) cannot distinguish spatial distributions due to the complex microstructures and various compositional materials of electrodes, identifying material constituents and phase transition in the microstructure of electrode coating from low contrast imaging (i.e., X-ray microtomography) becomes challenging.

Fig. 1(a) shows an example of a low contrast micro-scale image of an electrode. Every pixel of the image either contains carbon binder (C) or active particles, e.g., nickel (Ni), or a pore. Clearly, it is impossible to distinguish phase transitions between two material constituents by the human eye. Thus, we leverage the power of machine learning techniques to address the problem of material phase prediction from micro-scale images. Specifically, we develop a framework to label each pixel in an image with its material phase (i.e., C, Ni, or Pore). We treat this problem of pixel-wise material phase identification as an image segmentation task. Popular image segmentation architectures, (e.g., *U-Net* Ronneberger et al. [2015]), do not translate well to the current task. Fig. 1(b) shows the predicted material constituents for each pixel obtained by *U-Net*. Although U-Net effectively captures the global properties of material constituents, it fails to predict the phase boundaries (i.e., transition between two constituents) accurately. Additionally, the model predictions from U-Net also lead to *artifacts* (i.e., small regions of discontinuous predictions) especially at the phase boundary (see Fig. 1(b) - red rectangles are used to highlight artifacts). Accurate prediction of these constituents at phase boundaries is crucial as this influences the quantification of various physical properties of the electrode by domain experts. Even a minor error in predicting these phase boundaries translates to a high error in calculating the physical properties of the material.

We posit that the failure of traditional segmentation models to represent the microstructure accurately is two-fold: (a) the inability of *flat* modeling architectures to learn at different granularity across different regions of the microstructure. For example, predictions conditioned upon a coarse-grained representation might suffice in a homogeneous region of the microstructure. In contrast, a less homogeneous region (i.e., a region concentrated with many constituents) requires a relatively more fine-grained representation learning ability (b) traditional segmentation models do not possess the ability to accurately distinguish between tasks of differing degrees of difficulty (e.g., predicting pixels at phase transitions vs. predicting pixels within a single phase).

To address these failure modes, we propose a novel deep-learning framework, Material Phase Prediction (MatPhase), for LIB microstructure reconstruction. To address failure mode (a), we introduce two architectures that are designed to learn representations at varying complexities (i.e., a coarse-grained model and a fine-grained model). To address the failure mode (b), we develop a hierarchical curriculum learning framework to automatically distinguish between regions with low and high degrees of prediction difficulty. Fig. 1(c) shows our final predicted result after incorporating curriculum learning. Comparing with Fig. 1(b), see the pixels within corresponding red rectangles like the leftmost red rectangle. The small black dot of pore is not seen in the corresponding rectangle in Fig. 1(c). Overall, our predictions are smoother around phase boundaries without the artifacts that are apparent without using the curriculum.

Our Contribution: To the best of our knowledge, there is no existing technique to identify the complex phase boundaries from low contrast tomography images of electrodes, and MatPhase is the first attempt to address this challenge. Our contributions are: (i) We develop a novel uncertainty-aware ML framework MatPhase to predict material phases of electrodes from low-contrast micro-scale image samples obtained through X-ray tomography. (ii) First time in the literature, we introduce a *hierarchical curriculum learning* technique to predict the fine-grained material phase boundaries from a low-contrast image. (iii) To aid smooth fine-grained phase boundary prediction, we build a multi-class classifier which can capture both local and global representations. (iv) Through rigorous experiments, we show MatPhase yield better predictions than any non-trivial competitors.

2 Background and Preliminaries

We consider each individual material constituent in a low contrast micro-scale image of an electrode as a class and state our problem as a downstream task of material phase prediction.

Problem 2.1 (Material Phase Prediction)**.** ***Given***, *a set of labeled datasets* $\mathcal{D} = (\mathcal{X}, \mathcal{Y})_{i=1}^{N}$*, and a set of classes* $\mathcal{C}$*. Each* $\mathbf{X}_i \in \mathcal{X}$ *is an image of* $n \times m$ *pixels.* $\mathbf{Y}_i \in \mathcal{Y}$ *consists of* $n \times m$ *instances*[2] *for the corresponding pixels in* $\mathbf{X_i}$*, where each instance belongs to a class* $c \in C$*.* ***Predict***, $\hat{\mathbf{Y}}$ *for an unlabeled* $\mathbf{X}$.

Hierarchical Curriculum Learning: A learning task often requires a set of multiple curricula $\mathcal{S}$ (i.e., ordered learning strategies) to enable a model to learn effective representations. It is possible that these multiple curricula could each govern a subset of the overall dataset $\mathcal{D}$. Additionally, the set of curricula $\{s_1, .., s_n\} \in \mathcal{S}$ themselves have (latent) *dependencies*. In such a scenario, we introduce the paradigm of *hierarchical curriculum learning* (HCL) as a solution to address learning in the aforementioned context.

We define the goal of hierarchical curriculum learning (HCL) as being the implementation of multiple complementary curriculum strategies in a hierarchical fashion to order data by relative *hardness* (i.e., from easy to hard). Specifically, assume $\mathcal{S} = \{s_1, s_2\}$ where $|\mathcal{S}|$ (i.e., number of curricula) may be derived by expert knowledge. Further, let us assume $h(s_1) < h(s_2)$ where $h(\cdot)$ is a hardness measure also specified by a domain expert. s_1 may for instance be considered the curriculum geared toward learning global properties of the problem while s_2 may be the curriculum for learning (nuanced) local variations (thus $h(s_1) < h(s_2)$).

In this context, automatic-HCL comprises a learning framework which automatically learns to address the task of interest by first *partitioning* $\mathcal{D}$ into $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2\}$ such that $\mathcal{D}_1 \subseteq \mathcal{D}$, $\mathcal{D}_2 \subseteq \mathcal{D}$, $\mathcal{D}_1 \cap \mathcal{D}_2 = \emptyset$ and $h(\mathcal{D}_1) < h(\mathcal{D}_2)$. The partitioning step may be viewed as a *coarse-grained* curriculum that initially restricts learning to the easier instances in the dataset. This partitioning step not only learns effective representations for $\mathcal{D}_1$ but is also designed to yield *fine-grained* curricula for training models on the harder $\mathcal{D}_2$ data. Hence, the *coarse* and *fine-grained* curricula together comprise an automatic-HCL framework for effective learning of the task of interest.

3 Our Framework

This section describes our framework MatPhase, which predicts material phases from low-contrast images not easily perceived by humans. MatPhase first predicts the material phases through a global model (learning global data features). This is followed by a data partitioning step governed by the

[2] We employ the word ‘pixel’ and ‘instance’ interchangeably throughout the paper

performance of global model on $\mathcal{D}$, which forms the coarse-grained curriculum. The partitioning step concludes with the global model which learns the properties of easier data partition ($\mathcal{D}_1$), while yieliding a measure of hardness for $\mathcal{D}_2$, which forms the fine-grained curriculum. We design a local model to be trained on $\mathcal{D}_2$, governed by the fine-grained curriculum to learn local data features.

3.1 Uncertainty-aware Global Segmentation Model (UGSM)

We initially view Problem 2.1 as an image segmentation task using U-Net Ronneberger et al. [2015] due to its precise object localization and adaptability to be trained with a few samples. However, to achieve smooth prediction (as mentioned in Sec. 1 Fig. 1), we pursue U-Net as a global model instead. Our idea is to leverage a course-grained curriculum which learns the global representations of the model. Then, partition the incorrectly predicted instances (i.e., harder data $\mathcal{D}_2$) from the correctly predicted instances ($\mathcal{D}_1$). Hence, we aim for an uncertainty-aware global segmentation model (*UGSM*). Our assumption **A1** is: high uncertainty tends to incorrect predictions. We pursue Monte-Carlo Dropout (MCD) based uncertainty estimation Gal and Ghahramani [2016] due to its easy adaptability by any deep-learning model. For *UGSM*, we incorporate a dropout layer (dropout probability 0.5) after each convolution block of *U-Net* proposed by DeVries et al. DeVries and Taylor [2018]. During inference, *UGSM* samples T predictions. For each input $\mathbf{X}$, the final output of UGSM consists of two pairs: (i) predicted class per pixel, i.e, $\hat{\mathbf{Y}}_G \in \mathcal{R}^{n\times m}$, (ii) an uncertainty map $\mathbf{U}_G \in \mathcal{R}^{n\times m}$. Suppose, *UGSM* samples a set of logits $\mathcal{P} = \{\mathbf{P}^i\}_{i=1}^{T}$, where $\mathbf{P}^i \in \mathcal{R}^{|C|\times n\times m}$. Now if $\mathbf{F} = \frac{1}{T}\sum_{\mathbf{P}^i\in\mathcal{P}} \mathbf{P}^i$. $\hat{\mathbf{Y}}_G = \arg_{\max} \text{softmax}(\mathbf{F})$.

For uncertainty map $\mathbf{U}_G$, we compute the entropy each instance. Suppose, p_c^i is the probability that UGSM labels instance i as class c. We calculate $u^i = -\sum_{c\in C} p_c log(p_c)$.

3.2 Automatic Hierarchical Curriculum Learning (HCL)

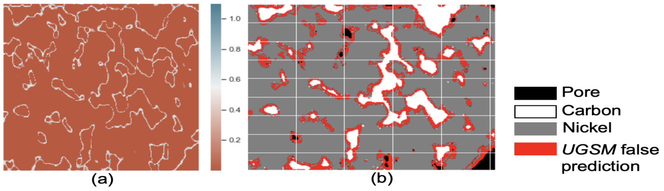

Figure 2: Motivating coarse-grained HCL: (a) Uncertainty map by UGSM (b) Heatmap of the predictions for same input of Fig. (a). Red pixels are incorrect predictions. All other pixels are correct predictions (black as pore, white as C, and grey as Ni). The density of red instances in Fig. (b) is higher than pixels with high uncertainty value in Fig. (a), indicating that UGSM also assigns low uncertainties to a significant portion of *incorrect* predictions invalidating assumption A1.

As mentioned in Sec. 2, our automatic *HCL* consists of coarse-grained curriculum s_1 and a fine-grained curriculum s_2. We view the incorrectly predicted instances of UGSM as harder data $\mathcal{D}_2$. In this context, our curriculum s_1 is the process of automatically partitioning $\mathcal{D}$ into $\mathcal{D}_1$ and $\mathcal{D}_2$, where $\mathcal{D}_1$ is the set of correctly predicted instances of UGSM. If assumption **A1** was to hold, s_1 could be a simple rule-based classifier with a pre-set uncertainty threshold that identifies $\mathcal{D}_2$ using the uncertainty map $\mathbf{U}_G$ obtained by UGSM. Fig. 2 (a) shows example of an uncertainty map of UGSM for an image $\mathbf{X}$. The high values (white colored pixels) denote high uncertainty. We observe that majority of the instances with high uncertainty lie at the *phase boundaries*. Fig. 2(b) shows the heatmap of the ground-truth label (material phases) for the same input $\mathbf{X}$. The red pixels represent *incorrect* predictions of UGSM. We notice that the number of incorrect classifications (Fig. 2(b) red pixels) is much higher than the number of instances with high uncertainty (Fig. 2(a)). This implies that UGSM assigns low uncertainties to a significant portion of instances that it predicts incorrectly. This implies that our simplistic assumption **A1** is a necessary but not a sufficient condition for identifying incorrect predictions in the context of the current model pipeline. Our goal through curriculum s_1 is hence to segregate (i.e., coarse-grained curriculum) all these red instances as $\mathcal{D}_2$.Next, we briefly describe the coarse-grained and fine-grained curriculum.

Coarse-grained curriculum (HCL-IDK): We intend s_1 to automatically learn the representations of $\mathcal{D}_1$ and $\mathcal{D}_2$ by UGSM for partitioning. Fig. 3 shows the heatmaps of the UGSM logits, $\mathbf{F}$ of each class $c \in \mathcal{C}$ (pore, carbon, and nickel) for the input $\mathbf{X}$ in Fig. 2. For e.g., consider Fig. 3(b) which is the heatmap of the logits of class carbon. The logits for all the pixels that contains class carbon (C) bear values within range $(2, 6)$. Also, the logits of the pixels across the C boundaries bear similar values, i.e., all values within range $(0.5, 1)$, (pixels colored in white). Comparing with Fig. 2(b), we see these white pixels or pixels of carbon boundaries (at Fig. 3(b)) match with the incorrectly predicted red pixels around carbon (Fig. 2(b)). We observe similar behavior for the heatmaps of logits of class pore and nickel. Hence, we can make an assumption **A2**: incorrectly predicted instances bear similar weights in the UGSM logits. For **A2**, we consider the incorrect instances as "I don't know" (*IDK*) class.

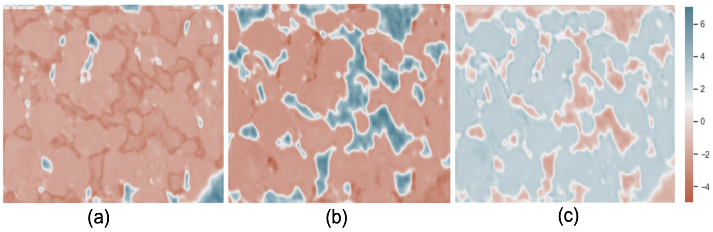

Figure 3: Heatmap of the logits of UGSM for the same input image of Fig. 2. Each Fig. corresponds to logits of class (a) Pore, (b) C, and (c) Ni. These heatmaps has high values (blue pixels) that corresponds to the predicted pixel class. Each heatmap also shows similar weights (pixels in white) around the class boundaries. Density of these white pixels mostly match with the incorrectly predicted red pixels in Fig. 2 (b). This motivates to view the incorrectly predicted pixels as "I dont Know" class.

We leverage a binary Feed-forward classifier for s_1 to identify the *IDK* instances. The goal of the classifier is to learn an effective latent representations for *IDK* through UGSM features, i.e. F and $\mathbf{U}_G$. For each instance j, the input of the classifier is a vector $\mathbf{idk}^j \in \mathcal{R}^4$ which consists of the features, i.e., $F^j \in \mathcal{R}^3$ and u^j. Let, $\mathcal{Y}_{IDK}$ be the set of all instances that classifier identifies as *IDK* and $\mathcal{Y}'_{IDK}$ be the complement of $\mathcal{Y}_{IDK}$. Hence, our easy data $\mathcal{D}_1$ comprises of the instances present in $\mathcal{Y}'_{IDK}$, and hard data $\mathcal{D}_2$ comprises of the instances in $\mathcal{Y}_{IDK}$.

Fine grained curriculum (HCL-USI): For learning the set of hard instances in $\mathcal{D}_2$, we adopt curriculum s_2. Here, s_2 is a fine-grained curriculum that quantifies the hardness of every instance in $\mathcal{D}_2$. We employ the uncertainty map $\mathbf{U}_G$ of UGSM as the hardness function. Hence, our s_2 curriculum progresses by training with increasingly hard instances governed by the instance-wise uncertainty of UGSM (HCL-USI).

3.3 Local Classification Model with Global Annotated Region (LCGAR)

The main goal of LCGAR is to provide smooth predictions for the *IDK* instances in $\mathcal{D}_2$. We aim to capture both the local and global features of the target instance for this task. We motivate for a classifier that accepts multiple features, i.e., (i) a local image of $\mathbf{X}$ surrounding the target *IDK* instance, (ii) predicted class labels of UGSM, i.e., $\hat{\mathbf{Y}}_G$, (iii) UGSM logits, i.e., $\mathbf{F}^j$ for the target instance j. By training our model with global features from UGSM, we intend to learn latent local representations of target instances, hence yield smooth predictions.

We denote $\mathbf{L}^j \in R^{h \times h}$ as bounded local region of the original input image $\mathbf{X}$ centered around the candidate pixel of interest j (target IDK instance). Here h is the width/height of the bounded region from the center of the pixel of interest. For the corresponding pixels in $\mathbf{L}^j$, we incorporate the predicted class of UGSM in the form of one hot encoder channel, i.e., a 3d input tensor ($\mathbf{I}_i^j \in \mathcal{R}^{|C| \times h \times h}$). If a predicted class of $\mathbf{L}_i^j$ is *IDK* by HCL-IDK, we interpolate $\mathbf{I}_i^j$ with zero masking.

We adopt a CNN architecture of LCGAR. To capture rich information about the arrangement of the surrounding for the target pixel of interest, we concatenate UGSM logits (F^j) with the flattening layer of CNN. The overall CNN architecture is shown in the appendix.

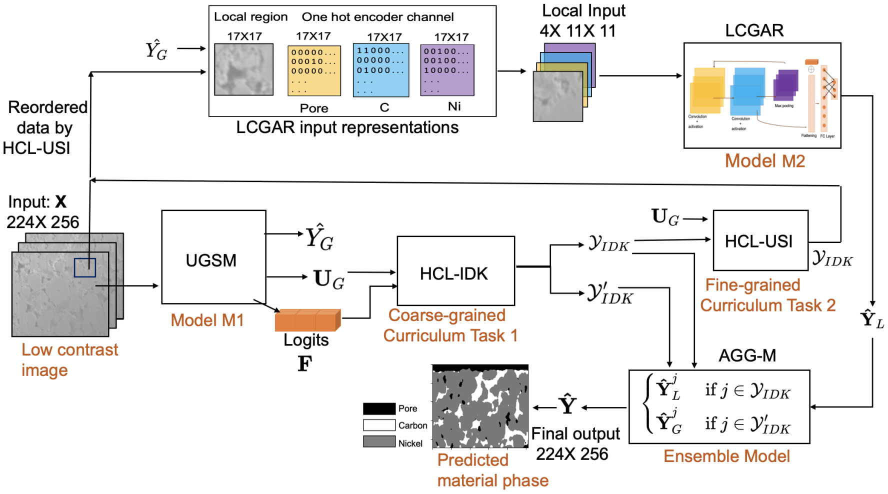

Figure 4: Overview of our proposed framework MatPhase which is designed to capture both global and local representations using automatic HCL-IDK and HCL-USI, for smooth phase predictions.

3.4 MatPhase Framework

Fig. 4 provides an overview of MatPhase. We first train the uncertainty-aware global segmentation model (UGSM). For each image, $\mathbf{X} \in \mathcal{X}$, UGSM provides a pixel-wise prediction map ($\hat{\mathbf{Y}}_G$) and an uncertainty map ($\mathbf{U}_G$) associated with the prediction. Our coarse-grained curriculum HCL-IDK is trained to partition all the instances in $\hat{\mathbf{Y}}_G$ into two sets, $\mathcal{Y}_{IDK}$ and $\mathcal{Y}'_{IDK}$. Next, the fine-grained curriculum HCL-USI orders $\mathcal{Y}_{IDK}$ by increasing prediction uncertainty obtained from the UGSM model. Next, our local model LCGAR is trained on both the global and local features to learn local context of $\mathcal{Y}_{IDK}$ to yield corresponding predictions $\hat{\mathbf{Y}}_L$. Finally, an ensemble model aggregates $\hat{\mathbf{Y}}_G$ and $\hat{\mathbf{Y}}_L$ to provide spatially smoother predictions, i.e., $\hat{\mathbf{Y}}$. $\hat{\mathbf{Y}} = \hat{\mathbf{Y}}^j_G$, if $j \in \mathcal{Y}_{IDK}$, else $\hat{\mathbf{Y}} = \hat{\mathbf{Y}}^j_L$.

4 Experiments

Our framework is implemented in PyTorch. All models are trained using 3 V100 Tesla GPUs. For every X-Ray Computed Tomography (XCT) imaging of electrode cross-sections, we obtain the ground-truth labels by utilizing cross-sectioning with Focused-Ion beam (FIB/SEM) experiments Vierrath et al. [2015]. The detail of the data collection process and pre-processing is in the appendix. We will release the code and datasets in public for research purposes.

Research Questions: Our goal is to demonstrate that MatPhase can capture better predictions than any other state-of-the-art (SOTA) image segmentation models. We are interested to analyze whether our automatic HCL framework is useful for modeling smooth phase predictions. Specifically we want to address: **Q1.** Is our automatic curriculum learning (HCL) effective for modeling phase boundaries? **Q2.** Does incorporating global information aid local modeling? **Q3.** How does MatPhase predict over SOTA? **Q1** and **Q2** are aligned with justifying the effectiveness of MatPhase. With **Q3**, we plan to showcase MatPhase performance comparing with multiple non-trivial competitors.

Measure of success: We evaluate the quality of MatPhase predictions on a downstream task of Problem 2.1 for low contrast image data. We choose four evaluation metrics: (i) F1-score. (ii) *Pixel accuracy (ACC)* for k best performing and worst performing predictions to evaluate smoothness of material phase predictions. (iii) *Mean intersection over union (mIU)* from the standard practice of image segmentation models Long et al. [2015], Badrinarayanan et al. [2017]. (iv) *Frequency weighted intersection over union (fIU)* to evaluate in presence of class imbalance from the standard practice for image segmentation models Long et al. [2015]. We describe each metric in the appendix.

Baselines: We compare MatPhase with SOTA image segmentation models. Specifically, to provide pixel to pixel predictions and to compare with different variants of UGSM, we use (1) DeepLabV3 Florian and Adam [2017] (2) MANet Fan et al. [2020](3) *FCN* Long et al. [2015], (4) *SegNet* Badrinarayanan et al. [2017], (5) *U-Net* Ronneberger et al. [2015], (6) *U-Net++* Zhou et al. [2018]. (7) *MCD-U-Net* DeVries and Taylor [2018]: Monte-Carlo dropout based U-Net, similar to our global model UGSM. (8) *Local-U-Net*: Replace our local model LCGAR with a U-Net model Ronneberger et al. [2015]. For training, we mask the cross-entropy loss values (with zeros) for the instances that are not classified by HCL-IDK. (9) *ResNet-18* He et al. [2016]: Our aim is to check with a pixel by pixel classifier model using same the input representation of LCGAR architecture. We leverage a *ResNet-18* model, setting the bounded local region $h = 11$ for every instance in the image. Note that, we also try with ResNet-34, ResNet-50 and only show the record which yield the best result. (10) *Adapted-LCGAR*: Predict every instances of image with pre-trained LCGAR.

4.1 Effectiveness (Q1-Q2)

Next, we aim to demonstrate the effectiveness of MatPhase, by analyzing the importance of (i) coarse-grained (HCL-IDK) and fine-grained (HCL-USI) curriculum of HCL, (ii) input representations of LCGAR, which incorporates global data features of to aid local modeling.

Table 1: Effectiveness of HCL for different variants of MatPhase in terms of F1-score per class, accuracy $(\%)$ on top 5 best performing, and lowest performing test sets, *mIU*, and *fIU*. For all metrics, high score yields the better result (best values in bold).

Model	F1			ACC(%)	ACC(%)	mIU	fIU
	Pore	C	Ni	Best 5	Lowest 5		
UGSM	**0.86**	0.82	0.94	91.5 ± 0.04	90.5± 0.6	0.77	0.844
MatPhase w/o HCL-USI	**0.86**	**0.85**	**0.95**	**93.2**± .07	91.7± 1.7	**0.803**	0.870
MatPhase	**0.86**	**0.85**	**0.95**	**93.2**± 0.02	**91.9**± 1.3	0.802	**0.871**

Effectiveness of HCL: For **Q1**, we want to evaluate the importance of curriculum learning. Hence, we describe the models varying the implementation of HCL learning strategies: (i) *UGSM* : We obtain the predictions from our pre-trained UGSM without any partitioning steps governed by curriculum HCL-IDK. (ii) *MatPhase w/o HCL-USI* : The model is trained directly using the hard data partitioned by HCL-IDK, which further is not learnt by HCL-USI. For testing, we replace our local model with *MatPhase w/o HCL-USI* and use the same framework MatPhase. We compare the performance of MatPhase, which is governed by both the curriculum HCL-IDK and HCL-USI against the models mentioned above in Table 1. We evaluate multiple metrics, mean F1-score per class across all the test datasets, average pixel accuracy rate of best 5 and worst 5 predictions, and mean IoU. For accuracy, the values (±) denote the percentage of deviation among the 5 predictions. We observe, MatPhase outperforms *UGSM* across all the metrics.

Effectiveness of UGSM features for LCGAR: For **Q2**, our goal is to capture how incorporating global features of UGSM to LCGAR aid modeling predictions. We use different variants of LCGAR varying the input representations. We train each model based on the same target instances used in MatPhase, only changing the input representation. (i) *Only-Image*: Consider LCGAR input as the local input image surrounding the target instances as mentioned in Sec. 3.3. We do not consider UGSM logits $\mathbf{F}^j$ in the flattening layer. (ii) *Image+UGSM-Y*: LCGAR input consists of the input images and the UGSM predictions $\hat{\mathbf{Y}}_G$ surrounding the target instances as mentioned in Sec. 3.3. We do not consider UGSM logits $\mathbf{F}^j$. (iii) *Image+UGSM-Emb*: Consider LCGAR input as the local input image surrounding the target instances. Also include logits $\mathbf{F}^j$ of the target instance j.

Table 2 shows the performance of LCGAR for different input representations. MatPhase uses all the global features as mentioned in Sec. 3.3. MatPhase clearly outperforms all the models. On average using global features, F1- score of MatPhase increase to 0.12 (0.17 for pore, 0.15 for C, and 0.03 for Ni) comparing with *Only-Image* which uses no global features. Comparing with *Image+UGSM-Y*, the low *mIU* and *fIU* indicates that MatPhase predicts better phase boundaries, hence lead to smooth prediction. Note that, all the models in Table 2 follow the exact same architecture of MatPhase, with the only difference being the change of input to LCGAR (i.e., the local component of the MatPhase). Details about each variant are described in Sec. 4.1.

Table 2: Effectiveness of incorporating global information for modeling LCGAR. Evaluation metrics are analogous to Table 1. MatPhase outperforms all the models (best values in bold).

Model	F1			ACC (%)	ACC (%)	mIU	fIU
	Pore	C	Ni	Best 5	Lowest 5		
Only-Image	0.69	0.7	0.92	87.6± 0.08	82.2± 1.9	0.65	0.781
Image+UGSM-Y	0.84	0.84	**0.95**	**93.3**± 0.06	89.8± 0.028	0.79	0.867
Image+UGSM-Emb	0.69	0.7	0.92	87.6± 0.08	82.2± 0.019	0.65	0.781
MatPhase	**0.86**	**0.85**	**0.95**	93.2± 0.02	**91.9**± 1.3	**0.802**	**0.871**

4.2 Comparison with baselines (Q3)

We show the performance of MatPhase against SOTA image segmentation models and some non-trivial competitors. Our goal is to demonstrate that achieving smooth predictions for material phases from low-contrast images is challenging and cannot be easily captured by a SOTA.

Table 3: Performance of MatPhase comparing with the baselines in terms of evaluation metrics analogous to Table 1. High value yields the better result for all the metrics (best values in bold).

Model	F1			ACC (%)	ACC (%)	mIU	fIU
	Pore	C	Ni	Best 5	Lowest 5		
DeepLabV3 Florian and Adam [2017]	0.85	**0.86**	0.87	90± 3.85	89± 1.87	0.76	0.83
MANet Fan et al. [2020]	0.81	0.79	0.93	90.7± 3.15	85.6± .0002	0.74	0.82
FCN Long et al. [2015]	0.66	0.17	0.88	80± 0.2	75.8± 1.2	0.45	0.64
SegNet Badrinarayanan et al. [2017]	0.81	0.78	0.94	91.7± 0.1	85.6± 1.6	0.74	0.83
U-Net Ronneberger et al. [2015]	0.85	0.82	0.94	91.2± .09	90.03± 1	0.77	0.84
U-Net++ Zhou et al. [2018]	0.76	0.53	0.78	72.4± 1.46	67.8± 3.09	0.54	0.59
MCD U-Net DeVries and Taylor [2018]	**0.86**	0.82	0.94	91.5± 0.09	90.5± 0.6	0.78	0.84
Local U-Net	**0.86**	0.82	0.94	91.5± 0.04	90.5± 0.5	0.78	0.84
ResNet-18 He et al. [2016]	**0.86**	0.83	**0.95**	92.7± .03	90.6± 0.02	0.79	0.86
Adapted-LCGAR	0.85	0.84	**0.95**	93± 0.02	91.8± 1.2	0.79	0.86
MatPhase	**0.86**	0.85	**0.95**	**93.2** ± 0.02	**91.9**± 1.3	**0.80**	**0.87**

Table 3 shows the performance MatPhase comparing with all the baselines described in Sec. 4. We observe MatPhase outperforms all the baselines. On average, our F1-score improves 0.5 for pore, 0.14 for C, and 0.2 for Ni. w.r.t. *mIU* and *fIU*, on average MatPhase performance improves 8.5% and 6% than the baselines. It indicates that leveraging automated curriculum learning (HCL) and incorporating global information to capture local and global features for predicting target instances can provide quality predictions (smooth and free of *artifacts*).

5 Conclusion

With the collaboration between computer scientists and battery researchers, for the first time, we present MatPhase, a novel framework to predict material phases, especially phase boundaries, from low contrast images of an electrode. Our extensive study against multiple non-trivial competitors (deep-learning-based models) and SOTA show that, on average, MatPhase increases the performance by 8.5%. We also show that incorporating HCL and the local model together on average increases our performance up to 7%[3]. Our framework is practically useful to battery researchers for understanding of the distribution of material phases of the electrode from high noise data. We envision using the framework to study the cross-sections of cycled electrodes and understand various degradation mechanisms that impact the loss of capacities.

[3] Best score difference among all metrics

References

V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

S. R. Daemi, C. Tan, T. Volkenandt, S. J. Cooper, A. Palacios-Padros, J. Cookson, D. J. Brett, and P. R. Shearing. Visualizing the carbon binder phase of battery electrodes in three dimensions. *ACS Applied Energy Materials*, 1(8):3702–3710, 2018.

T. DeVries and G. W. Taylor. Leveraging uncertainty estimates for predicting segmentation quality. *arXiv preprint arXiv:1807.00502*, 2018.

T. Fan, G. Wang, Y. Li, and H. Wang. Ma-net: A multi-scale attention network for liver and tumor segmentation. *IEEE Access*, 8:179656–179665, 2020.

L.-C. Florian and S. H. Adam. Rethinking atrous convolution for semantic image segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR). IEEE/CVF*, volume 6, 2017.

Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016.

P. Goyal and S. Ghosh. Hierarchical class-based curriculum loss. In *Porceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

M. Howard, M. C. Hock, B. Meehan, and L. Dresselhaus-Cooper. A locally adapting technique for boundary detection using image segmentation. *arXiv preprint arXiv:1707.09030*, 2017.

Z. Jiang, J. Li, Y. Yang, L. Mu, C. Wei, X. Yu, P. Pianetta, K. Zhao, P. Cloetens, F. Lin, et al. Machine-learning-revealed statistics of the particle-carbon/binder detachment in lithium-ion battery cathodes. *Nature communications*, 11(1):1–9, 2020.

T. M. LaBonte, C. Martinez, and S. A. Roberts. We know where we don't know: 3d bayesian cnns for uncertainty quantification of binary segmentations for material simulations. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2020.

Z. Liu, V. Manh, X. Yang, X. Huang, K. Lekadir, V. Campello, N. Ravikumar, A. F. Frangi, and D. Ni. Style curriculum learning for robust medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 451–460. Springer, 2021.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

P. Pietsch, M. Ebner, F. Marone, M. Stampanoni, and V. Wood. Determining the uncertainty in microstructural parameters extracted from tomographic data. *Sustainable Energy & Fuels*, 2(3): 598–605, 2018.

O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015.

S. Sinha, A. Garg, and H. Larochelle. Curriculum by smoothing. *Proceedings of Neural Information Processing Systems (NeuRIPS)*, 2020.

S. Vierrath, L. Zielke, R. Moroni, A. Mondon, D. R. Wheeler, R. Zengerle, and S. Thiele. Morphology of nanoporous carbon-binder domains in li-ion batteries—a fib-sem study. *Electrochemistry Communications*, 60:176–179, 2015.

X. Wang, Y. Luo, D. Crankshaw, A. Tumanov, F. Yu, and J. E. Gonzalez. Idk cascades: Fast deep learning by learning not to overthink. *arXiv preprint arXiv:1706.00885*, 2017.

X. Wang, Y. Chen, and W. Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang. Unet++: A nested u-net architecture for medical image segmentation. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 3–11. Springer, 2018.

6 Appendix

6.1 Relevant Work

Recently, multiple deep-learning based image segmentation models have been proposed for object or boundary detection in computer vision applications He et al. [2017], Ronneberger et al. [2015], Badrinarayanan et al. [2017], Long et al. [2015]. Howard et al. proposed a boundary detection model by addressing the spatially varying intensities in velocity shock wave image data Howard et al. [2017]. Researchers viewed the problem of identifying material constituents from electrode samples as object detection Daemi et al. [2018]. Errors induced during data binarization and how that would translate into uncertainties in the calculated parameters, such as porosity, tortuosity, or specific surface area was demonstrated by Pietsch et al. Pietsch et al. [2018]. Also, Jiang et al. developed a machine learning workflow that automatically identifies multiple fragments that broke away from the same particle and quantifies every single NMC particle in electrode Jiang et al. [2020]. Using this approach, they have also observed several regions of the NMC particles' that are detached from the polymeric binder domain in the severely damaged local region. Muller et al. adopted a 3D U-Net architecture to achieve segmentation across electrode samples Ronneberger et al. [2015]. However, none of these works adopt any non-trivial technique to achieve smooth predictions. Labonte et al. proposed a binary segmentation for LIB electrode material, which can quantify uncertainty to the segmentation LaBonte et al. [2020]. Multiple curriculum learning strategies have been proposed to solve different tasks Wang et al. [2021], Sinha et al. [2020]. Goyal et al. proposed a hierarchical class-based curriculum loss function to perform multi-class classification Goyal and Ghosh [2021]. To accelerate inference along with prediction accuracy, Wang et al. proposed a framework 'I don't know' (IDK) that systematically selects a subset of instances of a pre-trained deep learning model Wang et al. [2017]. Their proposed IDK framework is a rule-based function that selects instances if the entropy loss of the pre-trained model is within a user-defined threshold.

Liu et al. Liu et al. [2021] employs an external style transfer module for image segmentation which is used to govern the curriculum. However, for our method no external module is employed and curriculum is learnt implicitly as part of the framework MatPhase. To the best of our knowledge, our proposed hierarchical curriculum learning framework has not been adopted in any application for the task of material phase prediction.

6.2 Extension to Sec 3

LCGAR Architecture: Table 4 shows the architecture of LCGAR– a CNN. After each convolution and fully connected layer, we implement activation LeakyRelu. To capture rich information about the arrangement of the surrounding for the target pixel of interest, we concatenate UGSM logits (F^j) after flattening.

6.3 Extension to Sec 4

Data Collection Process: X-Ray Computed Tomography (XCT) and similar techniques do not provide information on carbon binder distribution as it is "transparent" for the X-Ray. It thus becomes indistinguishable from the pores in the electrodes. For collecting the ground-truth labels, to identify all the material constituents, we utilized cross-sectioning with Focused-Ion beam (FIB/SEM) experiments for imaging and Energy Dispersive X-Ray Spectroscopy (EDS) for chemical mapping of electrode cross-sections Vierrath et al. [2015]. The slicing was performed using Hitachi NB500 dual-beam FIB/SEM. The cross-sections were at a distance of every $200nm$ resulting in a total depth of $26\mu m$ resulting in 133 images.

Pre-processing: In this paper, we construct our data-set $\mathcal{D}$ using the corpus of 133 images collected through cross-sections of an electrode. Each low-contrast image $\mathbf{X}$ in the corpus consists of 224×256 pixels. The corresponding ground-truth (GT) $\mathbf{Y}$ contains 224×256 material constituents consisting of pores, carbon binder (C), and Nickel (Ni). We leverage a data augmentation technique for efficient training with few data samples. Our technique is: First, for every image $\mathbf{X}$ and the corresponding GT, we obtain k different images, removing a row of pixels from the top. Next, we resize each k images

Table 4: **LCGAR Architecture.**

Layer Type	**Specifications**
Convolution layer	Kernel (3×3,), 32 channels, padding 1
Convolution layer	Kernel (3×3,), 32 channels, padding 1
Maxpool	Kernel (2×2)
Convolution layer	Kernel (3×3,), 16 channels, padding 1
Convolution layer	Kernel (3×3,), 16 channels, padding 1
Maxpool	Kernel (2×2)
Convolution layer	Kernel (1×1,), 8 channels
Convolution layer	Kernel (1×1,), 8 channels
Concat: Flatten + UGSM logits $\mathbf{F}^j$	
Fully Connected Layer + dropout	Size 128, dropout (20%)
Batch-normalization	Size 128
Softmax	Number of class (3)

into original size, i.e., 224×256. We choose $k = 10$, as this is sufficient for training data. Finally, to smooth the resized GT for the corresponding k images, we use existing knowledge provided by domain experts: A pixel with Carbon (C) can not exist surrounding Ni pixels and vice-versa for a pixel with Ni. Using k nearest-neighbor rule, we remove noise from GT, i.e., if most surrounding pixels (nearest neighbors) around a C pixel are Ni, the GT label is changed to Ni. Thus, our labeled dataset $\mathcal{D}$ consists of $N = 1330$ images and corresponding GT. For the test, we split $\mathcal{D}$, which consists of 50 image-GT pairs. 40 pairs among the test-set are completely unseen samples, i.e., the original image of the electrode before the augmentation is not present in the training data.

Model Training: Both UGSM and LCGAR are trained with weighted cross-entropy loss due to heavy class imbalance in the ground-truth (on average C: 26%, Ni: 60%, Pore: 14%). In all the experiments, for LCGAR, we normalize input image between [-1, 1] and use activation *LeakyReLU*, which yields the best results. Our UGSM model is trained on 100 epochs with batch size 20, while HCL-IDK and LCGAR were trained for 15 epochs with batch size 1024.

Measure of Success:

(i) *F1-score*: Our goal is to measure the overall prediction for each class c, i.e., pore, carbon (C), and nickel (Ni) for the unseen datasets.

(ii) *Pixel accuracy (ACC)*: Fraction of the number of pixels that are predicted correctly among total pixels (in %).To measure smooth predictions Sinha et al. [2020], We aim to evaluate pixel accuracy for k best performing and worst performing predictions to evaluate smoothness of material phase predictions.

(iii) *Mean intersection over union (mIU)*: We intend to quantify predictions from standard practice of image segmentation models Long et al. [2015], Badrinarayanan et al. [2017]. Suppose, t_c be the total number of pixels labeled as class c, n_{jc} be the number of pixels of class c predicted as class j, and $|C|$ are the total number of classes. mIU $= \frac{1}{|C|} \sum_c \frac{n_{cc}}{t_c + \sum_{j \neq c} n_{jc}}$

(iv) *Frequency weighted intersection over union (fIU)*: To quantify predictions in presence of class imbalance, we incorporate *fIU* from the standard practice for image segmentation models Long et al. [2015]. If t be the total number of pixels, fIU $= \frac{1}{t} \sum_c \frac{t_c * n_{cc}}{t_c + \sum_{j \neq c} n_{jc}}$

6.3.1 Additional Experiments on Sensitivity of LCGAR

To check the effect of choosing different bounded local regions surrounding the target instance for LCGAR, we plan to vary the local bounded region of input (h). We use the same architecture of LCGAR, varying $h = x$, where x denotes the height and width of the local input region. We analyze $x = [11, 17, 25]$ which yield the best results. Table 5 shows the prediction performance on test datasets varying the size of the input region and complexity of the local model. We observe *LCGAR-h_{11}* which is built following the same architecture as our original LCGAR, outperforms all

Table 5: **Sensitivity analysis of LCGAR varying model parameters. *LCGAR-*h_{11} is our original LCGAR model. Evaluation metrics are analogous to Table 1. Our model *LCGAR-*h_{11} outperforms all the models (best values in bold).**

Model	**F1 score**	**ACC** (%) **Best 5**	**ACC**(%) **Lowest 5**	**mIU**	**fIU**
LCGAR$-h_{17}$	Pore: **0.86** C: 0.84 Ni: 0.94	**93.2** ± .04	91.7 ± 1.4	0.799	0.869
LCGAR$-h_{25}$	Pore: 0.85 C: **0.85** Ni: **0.95**	93 ± .01	91.8 ± 1.3	0.796	0.867
LCGAR$-h_{11}$	Pore: **.86** C: **0.85** Ni: **0.95**	**93.2** ± 0.02	**91.9** ± 1.3	**0.802**	**0.871**

other models across all the evaluation metrics (0.005 on average w.r.t. *mean IoU* and $.003$ on average w.r.t. *fIU*).

6.4 Real-world Impact

The ultimate goal of this work is to develop a deep learning model that can identify and distinguish multiple heterogeneous material phases in a 2D slice of battery electrodes from the experiments with feature sizes ranging in micrometers and having low contrast between different phases. The model would help quantify the confidence in phase transitions and contact areas between the phases from high noise data sets. It is known that binder phase morphologies significantly impact effective electrical conductivity and subsequently define the electro-chemically active area, pore phase tortuosity, and effective electrical conductivity. This knowledge would help advance the current understanding of binder distribution within the microstructure, which dictates the transport properties and help optimize the performance of the batteries. It is of paramount interest to understand the loss of contact between the binder phase and active material particles and the fracture of these particles after repeated cycling, which causes degradation and aging of batteries. We envision using MatPhase to study the cross-sections of cycled electrodes and understand various degradation mechanisms that impact the loss of capacities.