CONDITIONAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this work we tackle the problem of out-of-distribution generalization through conditional computation. Real-world applications often exhibit a larger distributional shift between training and test data than most datasets used in research. On the other hand, training data in such applications often comes with additional annotation. We propose a method for leveraging this extra information by using an auxiliary network that modulates activations of the main network. We show that this approach improves performance over a strong baseline on the Inria Aerial Image Labeling and the Tumor-Infiltrating Lymphocytes (TIL) Datasets, which by design evaluate out-of-distribution generalization in both semantic segmentation and image classification.

1 INTRODUCTION

Deep learning has achieved great success in many core artificial intelligence (AI) tasks (Hinton et al., 2012; Krizhevsky et al., 2012; Brown et al., 2020) over the past decade. This is often attributed to better computational resources (Brock et al., 2018) and large-scale datasets (Deng et al., 2009).

Collecting and annotating datasets which represent a sufficient diversity of real-world test scenarios for every task or domain is extremely expensive and time-consuming. Hence, sufficient training data may not always be available. Due to many factors of variation (e.g., weather, season, daytime, illumination, view angle, sensor, and image quality), there is often a distributional change or domain shift that can degrade performance in real-world applications (Shimodaira, 2000; Wang & Schneider, 2014; Chung et al., 2018). Applications in remote sensing, medical imaging, and Earth observation commonly suffer from distributional shifts resulting from atmospheric changes, seasonality, weather, use of different scanning sensors, different calibration and other variations which translate to unexpected behavior at test time (Zhu et al., 2017; Robinson et al., 2019; Ortiz et al., 2018).

In this work, we present a novel neural network architecture to increase robustness to distributional changes (See Figure 1). Our framework combines conditional computation (Dumoulin et al., 2018; 2016; De Vries et al., 2017; Perez et al., 2018) with a task specific neural architecture for better domain shift generalization.

One key feature of this architecture is the ability to exploit extra information, often available but seldom used by current models, through a conditioning network. This results in models with better generalization, better performance in both independent and identically distributed (i.i.d.) and non- i.i.d. settings, and in some cases faster convergence. We demonstrate these methodological innovations on an aerial building segmentation task, where test images are from different geographic areas than the ones seen during training (Maggiori et al., 2017) and on the task of Tumor Infiltrating Lymphocytes (TIL) classification (Saltz et al., 2018).

We summarize our main contributions as follows:

- We propose a novel architecture to effectively incorporate conditioning information, such as metadata.
- We show empirically that our conditional network improves performance in the task of semantic segmentation and image classification.
- We study how conditional networks improve generalization in the presence of distributional shift.

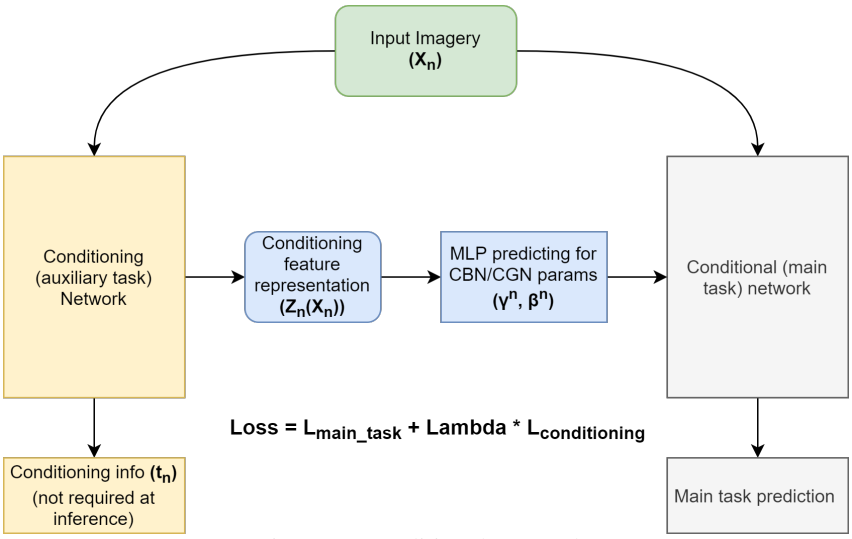


Figure 1: Conditional Networks

2 Background and Related Work

Self-supervised learning. Self-supervised learning extracts and uses available relevant context and embedded metadata as supervisory signals. It is a representation learning approach that exploits a variety of labels that come with the data for free. To leverage large amounts of unlabeled data, it is possible to set the learning objectives such that supervision is generated from the data itself. The self-supervised task, also known as pretext task, guides us to a supervised loss function (Gidaris et al., 2018; Oord et al., 2018; He et al., 2019; Chen et al., 2020). However, in self-supervised learning we usually do not emphasize performance on this auxiliary task. Rather we focus on the learned intermediate representation with the expectation that this representation can carry good semantic or structural meanings and can be beneficial to a variety of practical downstream tasks. Conditional networks can be seen as a self-supervision approach in which the pretext task is jointly learned with the downstream task.

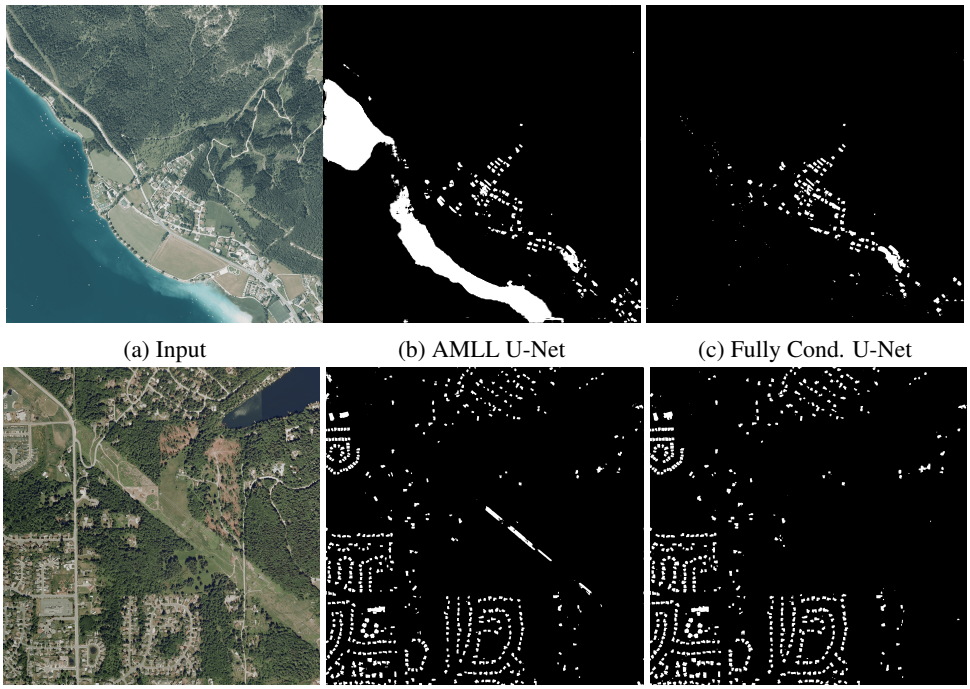
Our proposed modulation of a segmentation architecture based on an auxiliary network's intermediate representation can also be seen as an instance of *knowledge transfer* (Hinton et al., 2015; Urban et al., 2016; Buciluǎ et al., 2006). Because the auxiliary network has an additional task signal – metadata prediction – information about this task can be transferred to the segmentation network.

Conditional Computation. Ioffe and Szegedy designed Batch Normalization (BN) as a technique to accelerate the training of deep neural networks (Ioffe & Szegedy, 2015). BN normalizes the feature maps of a given mini-batch $B = \{F_{n,.,.,.}\}_{n=1}^{N}$ of N training samples as described by the following Equation:

$$BN(F_{n,c,h,w}|\gamma_c, \beta_c) = \gamma_c \frac{F_{n,c,h,w} - \mathbb{E}_B[F_{.,c,.,.}]}{\sqrt{\mathrm{Var}_B[F_{.,c,.,.}] + \epsilon}} + \beta_c, \tag{1}$$

where c, h and w are indexing the channel, height and width axis, respectively, γ_c and β_c are trainable scale and shift parameters, introduced to keep the representational power of the original network, and ϵ is a constant factor for numerical stability. For convolutional layers the mean and variance are computed over both the batch and spatial dimensions, implying that each location in the feature map is normalized in the same way.

De Vries et al. (2017); Perez et al. (2018) introduced Conditional Batch Normalization (CBN) as a method for language-vision tasks. Instead of setting γ_c and β_c in Equation 1 directly, CBN defines them as learned functions $\beta_{n,c} = \beta_c(q_n)$ and $\gamma_{n,c} = \gamma_c(q_n)$ of a conditioning input q_n. Note that this results in a different scale and shift for each sample in a mini-batch. Scale ($\gamma_{n,c}$) and shift ($\beta_{n,c}$) parameters for each convolutional feature are generated and applied to each feature via an affine transformation. Feature-wise transformations frequently have enough capacity to model complex



(a) Input (b) AMLL U-Net (c) Fully Cond. U-Net

(d) Input (e) AMLL U-Net (f) Fully Cond. U-Net

Figure 2: Qualitative Results. (a) Input from Tyrol East city, (b) AMLL U-Net segmentation results for (a), (c) Fully Cond. U-Net CGN segmentation results for (a), (d) Input from Bellingham city, (e) Fully Cond. U-Net CGN segmentation results for (d), (f) Fully Cond. U-Net CGN segmentation results for (d).

phenomena in various settings (Dumoulin et al., 2018). For instance, they have been successfully applied to neural style transfer (Dumoulin et al., 2016) and visual question answering (Perez et al., 2018). This kind of conditional computation scheme is not tied to the type of normalization used. Wu & He (2018) recently proposed Group Normalization (GN). GN divides feature maps into groups and normalizes the features within each group. GN only uses the layer dimension, hence its computation is independent of batch size. Ortiz et al. (2020) proposed Local Context Normalization (LCN) to encourage local contrast enhancement by normalizing features based on a spatial window around it and the filters in its group. Recently, Michalski et al. (2019) showed that Conditional Group Normalization (CGN) offer performance similar to CBN. In this work, we show results using CBN and the conditional form of GN known as CGN.

3 Formulation and Network Architecture

3.1 Problem Abstraction

We first establish notation. Let $(x_n)_{n=1}^N$ represent the training images. Each image x_n is associated with a corresponding label for the task of interest, y_n. From now on we refer to the task of interest as main task.

Available extra annotation is denoted by t_n. An auxiliary conditioning network learns an embedding z_n of the input image, which is used to predict t_n and to modulate layers of the main task network. There is flexibility in the choice of t_n. For example, in the building segmentation problem – revisited in Section 4.1 – t_n could be anything from the name of the neighborhood, its geographic coordinates or a flag indicating whether it is urban or rural.

Given a particular input image x_n, we adapt our segmentation mask y_n depending on the structure of z_n. Effectively, this allows the model to learn a family of regressors, parameterized by z_n's. Each member of this family is then free to adapt to its contextual z_n. The representations z_n must be learned and we propose to guide the learning by the auxiliary task of predicting the metadata

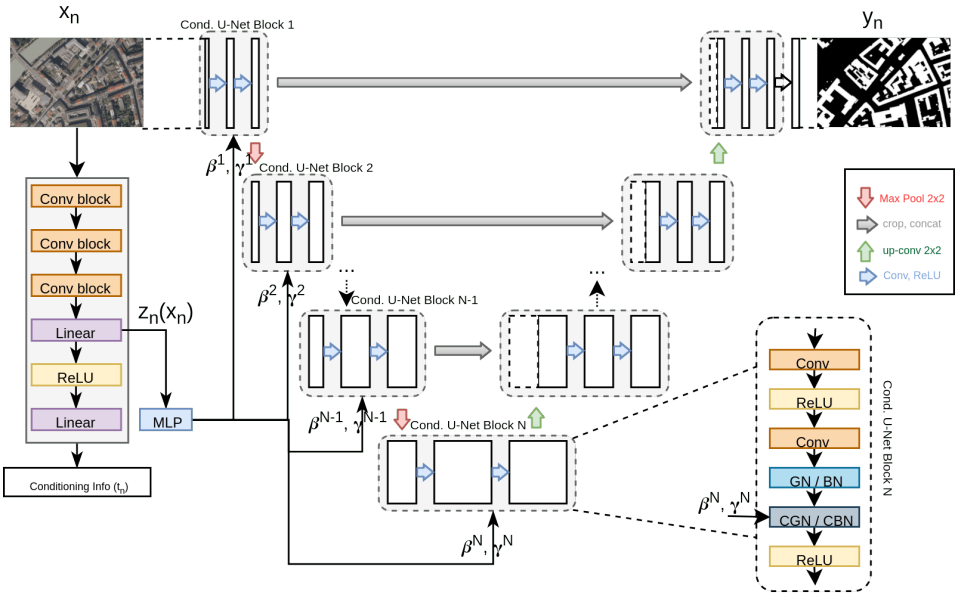


Figure 3: Conditional Network Architecture for U-Net

t_n. To encourage this, we train an auxiliary network to predict the metadata t_n given x_n and use intermediate high-level features $\hat{z}(x_n)$ as a proxy for z_n. This way of leveraging the additional information indirectly is an important piece of the proposed architecture, since using the raw t_n is not useful for generalization. Consider the example of building segmentation discussed in Section 4.1, where we use geographical coordinates as extra information. Imagine the model is presented with an image from a new city, which might have features that are visually similar to cities from the training set, but has very different coordinates. If we conditioned on the location alone, we would not be able to generalize to this new city. Another important property of leveraging metadata like this is that we do not need access to t_n at test time.

Learning $\hat{z}(x_n)$ in this way can give useful hints to the main task network. For example, if we know we are in a city (target label t_n) with a certain visual characteristic, such as predominantly smaller buildings, then we hypothesize that this will be captured by $\hat{z}(x_n)$. This yields a useful inductive bias for the segmentation task as the labeled regions (in the segmentation mask y_n) are probably smaller than usual. Of course, if the image x_n is clear, the model has enough capacity and we have enough training data, such a hint may not be necessary. However, as stated in the introduction, this is usually not feasible in practice, as any reasonably sized dataset cannot densely cover all possible variations. We hypothesize that while both the main task network and the auxiliary network only receive the image as input, the architecture and training mechanism used by conditional networks help to learn efficient representations that improve generalization.

3.2 Network Architecture

Our proposed architecture transforms any standard convolutional neural network into one that incorporates conditioning information t. After convolutional blocks in a neural network, we add a linear operation to scale and shift every feature map c as follows: $(\beta_c(\hat{z}(x)) \times c) + \gamma_c(\hat{z}(x))$. This affine transformation is applied after normalization operations of (sub-)batch or feature map statistics. We refer to our family of networks as Conditional Networks. Figure 3 shows this extension applied to the popular U-Net Ronneberger et al. (2015) architecture. U-Net is an encoder-decoder network architecture with skip connections. Figure 3 shows the auxiliary network on the left modulating the modified U-Net on the right. We obtain $\beta_{n,c}(\hat{z}(x_n))$ and $\gamma_{n,c}(\hat{z}(x_n))$ from the conditioning network. The conditioning network is a convolutional architecture (LeCun et al., 1998) followed by a fully-connected layer that predicts metadata t_n as a function of the input image x_n. The pre-activation features before the output layer are used as $\hat{z}(x_n)$. The functions $\beta_{n,c}(\hat{z}(x_n))$ and $\gamma_{n,c}(\hat{z}(x_n))$ mapping $\hat{z}(x_n)$ to the scale and shift parameters are implemented with an multi-

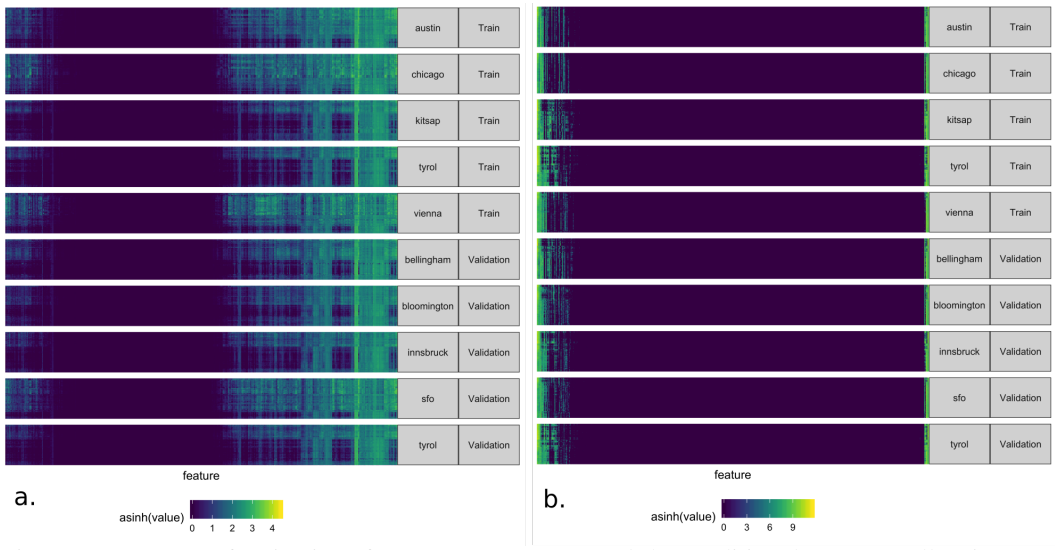


Figure 4: Heatmap of activations for (a) AMLL U-Net and (b) conditional U-Net. Cell ij in row i and column j gives the activation of features j on patch i. Rows and columns are sorted so that those that are more similar to one another appear side-by-side. Note that the color legends have different scales.

layer perceptron (MLP). Using the latent representations instead of directly using t_n allows us to leverage combinations of features that were useful in localizing images from previously seen data, potentially improving generalization.

Because all its parts are differentiable, conditional networks can be trained end-to-end using gradient-based optimization. Our full objective is described in Equation 2, where α is a hyper-parameter. $L_{\text{main_task}}$ represents a standard main task loss. $L_{\text{main_task}}$ depends on the task, such as Jaccard, cross-entropy, and dice for semantic segmentation. $L_{\text{conditioning}}$ ensures the conditioning networks correctly predicts t_n.

$$L_{\text{cond._net}} = L_{\text{main_task}} + \alpha * L_{\text{conditioning}} \tag{2}$$

4 Experiments

We study the following hypotheses:

H1: Generalization through context. Explicit incorporation of conditioning information improves generalization in semantic segmentation and image classification tasks.

H2: Interpretability. The features learned by the conditioning and the segmentation network reflect context-specific and context-invariant information, respectively.

4.1 Conditional Networks for Semantic Segmentation of Aerial Images

To study hypotheses H1 and H2 we focus on the Inria Aerial Image Labeling Dataset. This dataset was introduced to test out-of-distribution generalization of remote-sensing segmentation models (Maggiori et al., 2017). It includes imagery from 10 dissimilar urban areas in North America and Europe. Instead of splitting adjacent portions of the same images into training and test sets, the splitting was done city wise. All tiles of five cities were included in the training set and the remaining cities are used as test set. In this paper we refer to the test set as a *transfer set*. The transfer set also has variation in illumination, landscape, and time, making it well-suited to evaluate out-of-distribution generalization. The provided imagery comes orthorectified and has a spatial resolution of 0.3m per pixel covering $810km^2$ ($405km^2$ for training and $405km^2$ for the transfer set) on an evenly spaced grid. Images were labeled for the semantic classes of *building* and *not building*. We use geographical coordinates of the images as target data for the auxiliary network (see Section 4.1).

Table 1: Inria Transfer Set Performance.

Method	Bellingham		Bloomington		Innsbruck		San Francisco		East Tyrol		Overall	
	IoU	Acc.	IoU	Acc.	IoU	Acc.	IoU	Acc.	IoU	Acc.	IoU	Acc.
AMLL U-Net	65.37	**96.53**	55.07	95.83	67.62	96.08	72.80	91.00	67.00	96.91	67.98	95.27
U-Net + GN	55.48	93.38	55.47	94.41	58.93	93.77	72.12	89.56	62.27	95.73	63.71	93.45
U-Net + LCN	63.61	96.26	60.47	96.22	68.99	96.28	**75.01**	**91.46**	68.90	97.19	69.90	**95.48**
Cond. U-Net CBN	60.38	95.91	48.84	95.20	65.99	95.97	72.74	90.88	69.70	97.24	66.54	95.04
Cond. U-Net CGN	63.10	96.15	56.43	95.84	68.70	96.23	74.31	91.28	66.71	97.03	68.66	95.31
Fully Cond. U-Net CBN	65.91	96.49	57.24	95.95	63.14	95.71	73.89	90.83	69.24	97.21	68.50	95.24
Fully Cond. U-Net CGN	**66.98**	96.52	**63.27**	**96.33**	**69.80**	**96.24**	73.88	90.75	**70.77**	**97.35**	**70.55**	95.44
$nont_n$ Cond. U-Net	62.97	96.10	53.08	95.03	65.61	95.73	71.87	90.32	65.66	97.75	66.33	94.78

Table 2: Distribution Shift Generalization Gap.

Method	Val. set	Transfer set	Gen. Gap
	IoU (%)	IoU (%)	IoU (%)
AMLL U-Net	71.87	67.98	3.89
U-Net + GN	71.38	63.79	7.59
Cond. U-Net CBN	72.15	68.50	3.65
Cond. U-Net CGN	**72.77**	**70.55**	**2.22**

For H1, we compare model performances using the standard benchmark training-transfer split. For H2, we perform an exploratory visualization of the feature-activation maps for the different models.

Generalization via conditioning. We used the Inria standard train-transfer split to see whether conditioning information helps out-of-distribution generalization. From the training set we reserved five images of each city for the validation set as suggested by Maggiori et al. (2017).

For this set of experiments we trained our conditional U-Net presented in Figure 3 end-to-end from scratch. We used as segmentation network the AMLL U-Net as described by Huang et al. (2018), which is a version of U-Net with fewer filters. The AMLL U-Net was the winning entry and top of the Inria leaderboard and we use it as baseline for comparison in this section.

The Conditional U-Net was trained exactly as AMLL U-Net. We used the Adam optimizer (Kingma & Ba, 2014) with cross-entropy as segmentation loss $L_{\text{main_task}}$. We used an initial learning rate of 0.001. We trained our network for 100 epochs, where each epoch consisted of 8000 patches of size 572×572 obtained from the image tiles, and the learning rate was reduced to 0.0001 after 60 epochs exactly as for Duke AMLL U-Net, but without any data augmentation technique[1]. We used the standardized latitude and longitude of the center pixel of each patch as the conditioning information to be predicted by the conditioning network and the mean-squared error (MSE) as the conditioning loss $L_{\text{conditioning}}$. All training details and a figure showing histograms of final Intersection-over-Union (IoU) scores of the different models can be found in the SI.

Table 1 shows the performance of the AMLL U-Net baselines and different variations of our proposed architecture on the transfer set. Conditioning uniformly improved segmentation performance over the corresponding conditioning-free models. This empirically validates our hypothesis H1 about generalization through context. We identify as *Cond. U-Net* those models in which both the encoder and decoder are modulated, which yielded a small gain in performance over just modulating the encoder or decoder alone. We recommend conditioning all blocks since the extra computational cost is very small. Modulating using CGN consistently outperforms CBN.

t_n Matters. An experiment using the conditioning network without the auxiliary task of predicting t_n shows a deterioration of performance relative to the baseline U-Net as shown in Table 1. We see this as evidence for the importance of guiding the learning process of the latent conditioning representation $\hat{z}(x_n)$.

Table 2 shows the generalization gap as the difference between the validation set (i.i.d.) and the transfer set performance. Notice how the models' performances consistently degrade when we evaluate them on cities not seen during training. Conditioning substantially reduces the generalization

[1]The source code with details explaining how to run our experiments is available here: `https://xyz.ab`

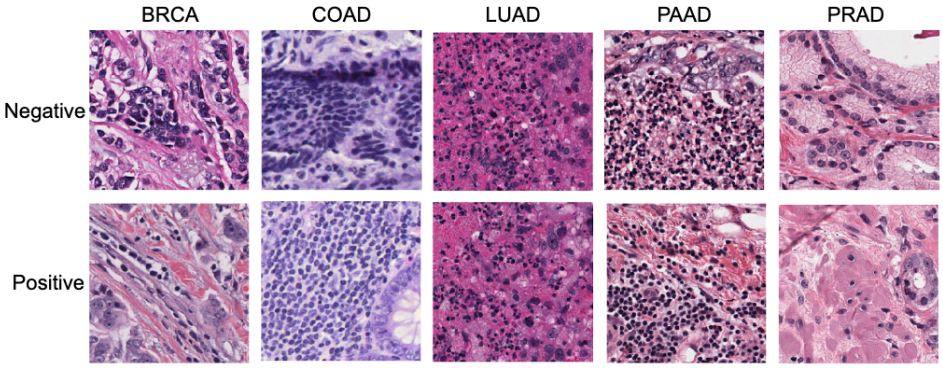


Figure 5: Example of negative and positive TIL patches for different cancer types. The desired task is to properly classify a patch as TIL negative or TIL positive independently of the cancer type.

gap induced by the distribution-shift between the training and transfer sets, yielding evidence for hypothesis H1.

Interpretation of conditioning features. To evaluate hypothesis H2, we analyze patterns of activations across these experiments. This hypothesis is interesting for several reasons,

- It serves as a sanity check for the proposed architecture, ensuring that supervision from conditioning information leads to features that do indeed distinguish between cities.
- It sheds light on the potential of using conditioning information to facilitate learning of generalizable features by intentionally learning context-dependent features.
- We can begin to characterize how generalization occurs, by identifying which training cities a test patch "looks" like.

4.2 Conditional Networks for Tumor Infiltrating Lymphocytes Classification

We also test Conditional Networks for the task of tumor infiltrating lymphocytes (TIL) Classification. During the cancer diagnosis and treatment process, a patient may have a biopsy, which produces a diagnostic tissue sample. Using this sample, a slide is prepared and examined under a microscope by a pathologist to understand both how to treat the disease and to provide a prognosis for the patient's future. Virtually all cancer patients undergo these biopsies, producing large volumes of these pathology slides.

A significant feature in these images is tumor infiltrating lymphocytes (TILs), which are types of immune cells that move into a tumor to try to attack the cancer. The quantification of TILs is well known to have prognostic value in many contexts (Fridman et al., 2012; Angell & Galon, 2013) because understanding patient immune response to tumors is becoming increasingly important with the growth of cancer immunotherapy. Features such as TILs can be quantified through image analysis and deep learning algorithms (Saltz et al., 2018; Klauschen et al., 2018). In Saltz et al. (2018), a convolutional neural network (CNN) architecture is systematically optimized to carry out classification of nuclei from pathology images. This led to the release of a dataset consisting of TIL maps corresponding to roughly 5,000 whole slide images from The Cancer Genome Atlas (TCGA) where the task is to classify TIL patches as positive or negative. The training set, we used 86,000 patches that were manually annotated with TIL classification (Saltz et al. (2018)). In the full set, there are 64,381 TIL negative patches and 21,773 TIL positive patches. All training patches used are in 100×100 pixel resolution, 20 times magnification, and are annotated as TIL positive or TIL negative. Examples of the training images and their labels are given in Figure 5.

These training images represent seven different cancer types: invasive carcinoma of the breast (BRCA), colon adenocarcinoma (COAD), lung adenocarcinoma (LUAD), pancreatic adenocarcinoma (PAAD), prostate adenocarcinoma (PRAD), skin cutaneous melanoma (SKCM), and endometrial carcinoma of the uterine corpua (UCEC). We use another 652 patches as our validation set (all from the LUAD cancer type), and 900 manually annotated patches from twelve cancer types in total as the testing set. The twelve cancer types are the seven listed above, as well as five novel ones

Table 3: TIL Classification Test (generalization set) Results

Model	AUC (%)
Spatial (Saltz et al. (2018); Hou et al. (2019))	86.16
Inception (Szegedy et al. (2016); Patton et al. (2019))	89.9
MENNDL (Patton et al. (2019))	86.16
Conditional VGG16 (ours)	**92.91**
Direct Cond. VGG16	50.16
Cond. VGG16 alpha 0	77.18

never seen during training (urothelial carcinoma of the bladder (BLCA), cervical squamous cell carcinoma and endocervical adenocarcinoma (CESC), lung squamous cell carcinoma (LUSC), rectal adenocarcinoma (READ), and stomach adenocarcinoma (STAD)).

Methods tested. We train the VGG 16-layers network as baseline (Simonyan & Zisserman (2014)). VGG networks have been shown to work well for pathology image classification (Xu et al. (2015); Hou et al. (2016)). We then build the conditional version of VGG16 which we refer to as Conditional VGG16. Conditional VGG16 is created in a similar way as conditional U-Net and share a similar conditioning network. Instead of doing regression the conditioning network is trained to perform image classification for the type of cancer (t_n). As with conditional U-Net, we predict $\beta_{n,c}(\hat{z}\,(x_n))$ and $\gamma_{n,c}(\hat{z}\,(x_n))$ from features in the conditioning network. $\beta_{n,c}(\hat{z}\,(x_n))$ and $\gamma_{n,c}(\hat{z}\,(x_n))$ are then used to modulate VGG16 architecture. We use binary cross-entropy as $L_{\text{main_task}}$ and multiclass cross-entropy as $L_{\text{conditioning}}$. Training details are provided in the supplemental material.

Results. Table 3 shows the results for the Tumor-infiltrating Lymphocyte classification task using different approaches. Augmenting VGG16 using conditional networks improves its performance by a large margin allowing us to obtain state-of-the-art performance in the task, even improving over previous top performing approaches. It is also important to remember that t_n (cancer type) is only required during training. Inference only requires the input image, as in competing methods.

To show the importance of the conditioning network we trained a version of conditional VGG16 where conditioning parameters are predicted directly from the cancer type one hot encoding ($\beta_{n,c}\,(t_n)$, $\gamma_{n,c}\,(t_n)$) and we refer to it as "direct cond. VGG16" on Table 3. As expected, doing so does not work well. Properly predicting for the conditioning task is also very important for conditional networks to work well. When we only optimize for $L_{\text{main_task}}$ performance degrades as it is shown on Table 3 under "Cond. VGG16 alpha 0".

5 Discussion and Conclusions

We have presented *conditional networks*, a family of neural networks that leverages conditional information for improved performance and better generalization. Conditional networks can be applied to any network architecture that uses normalization layers. We showed how the performance of two widely adopted network architectures (U-Net and VGG16) can be greatly improved by applying conditional networks for both semantic segmentation and image classification. We have shown that conditional networks consistently reduce the generalization gap observed when there is a shift of the underlying distribution at test time. After carefully studying the network feature activations, we found that the improved generalization ability of the proposed network is not due to the ability of learning more invariant features. It appears, instead, that the conditional network learns a smaller collection of features more relevant to the task. Conditional networks exploits available extra annotation t_n while training, however t_n is not required during inference. It is not always obvious which choice of t_n helps the most to better estimate z_n. Future work involves studying how the choice of t_n influences performance in a larger set of tasks.

References

Helen Angell and Jérôme Galon. From the immune contexture to the immunoscore: the role of prognostic and predictive immune markers in cancer. *Current opinion in immunology*, 25(2):

261–267, 2013.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541. ACM, 2006.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Yeounoh Chung, Peter J Haas, Eli Upfal, and Tim Kraska. Unknown examples & machine learning model generalization. *arXiv preprint arXiv:1808.08294*, 2018.

Harm De Vries, Florian Strub, Jérémie Mary, Hugo Larochelle, Olivier Pietquin, and Aaron C Courville. Modulating early visual processing by language. In *Advances in Neural Information Processing Systems*, pp. 6594–6604, 2017.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. *arXiv preprint arXiv:1610.07629*, 2016.

Vincent Dumoulin, Ethan Perez, Nathan Schucher, Florian Strub, Harm de Vries, Aaron Courville, and Yoshua Bengio. Feature-wise transformations. *Distill*, 2018. doi: undefined. https://distill.pub/2018/feature-wise-transformations.

Wolf Herman Fridman, Franck Pagès, Catherine Sautes-Fridman, and Jérôme Galon. The immune contexture in human tumours: impact on clinical outcome. *Nature Reviews Cancer*, 12(4):298–306, 2012.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *International Conference on Learning Representations (ICLR)*, 2018.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Le Hou, Kunal Singh, Dimitris Samaras, Tahsin M Kurc, Yi Gao, Roberta J Seidman, and Joel H Saltz. Automatic histopathology image analysis with cnns. In *2016 New York Scientific Data Summit (NYSDS)*, pp. 1–6. IEEE, 2016.

Le Hou, Vu Nguyen, Ariel B Kanevsky, Dimitris Samaras, Tahsin M Kurc, Tianhao Zhao, Rajarsi R Gupta, Yi Gao, Wenjin Chen, David Foran, et al. Sparse autoencoder for unsupervised nucleus detection and representation in histopathology images. *Pattern recognition*, 86:188–200, 2019.

Bohao Huang, Kangkang Lu, Nicolas Audeberr, Andrew Khalel, Yuliya Tarabalka, Jordan Malof, Alexandre Boulch, Bertr Le Saux, Leslie Collins, Kyle Bradbury, et al. Large-scale semantic classification: outcome of the first year of inria aerial image labeling benchmark. In *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, pp. 6947–6950. IEEE, 2018.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Preceedings of the International Conference in Machine Learning (ICML)*, 2015.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Frederick Klauschen, K-R Müller, Alexander Binder, Michael Bockmayr, M Hägele, P Seegerer, Stephan Wienert, Giancarlo Pruneri, S de Maria, S Badve, et al. Scoring of tumor-infiltrating lymphocytes: From visual estimation to machine learning. In *Seminars in cancer biology*, volume 52, pp. 151–157. Elsevier, 2018.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2017.

Vincent Michalski, Vikram Voleti, Samira Ebrahimi Kahou, Anthony Ortiz, Pascal Vincent, Chris Pal, and Doina Precup. An empirical study of batch normalization and group normalization in conditional computation. *arXiv preprint arXiv:1908.00061*, 2019.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Anthony Ortiz, Alonso Granados, Olac Fuentes, Christopher Kiekintveld, Dalton Rosario, and Zachary Bell. Integrated learning and feature selection for deep neural networks in multispectral images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1196–1205, 2018.

Anthony Ortiz, Caleb Robinson, Mahmudulla Hassan, Dan Morris, Olac Fuentes, Christopher Kiekintveld, and Nebojsa Jojic. Local context normalization: Revisiting local normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Robert M Patton, J Travis Johnston, Steven R Young, Catherine D Schuman, Thomas E Potok, Derek C Rose, Seung-Hwan Lim, Junghoon Chae, Le Hou, Shahira Abousamra, et al. Exascale deep learning to accelerate cancer research. *arXiv preprint arXiv:1909.12291*, 2019.

Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

C. Robinson, Le Hou, Kolya Malkin, Rachel Soobitsky, Jacob Czawlytko, Bistra Dilkina, and Nebojsa Jojic. Large scale high-resolution land cover mapping with multi-resolution data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241. Springer, 2015.

Joel Saltz, Rajarsi Gupta, Le Hou, Tahsin Kurc, Pankaj Singh, Vu Nguyen, Dimitris Samaras, Kenneth R Shroyer, Tianhao Zhao, Rebecca Batiste, et al. Spatial organization and molecular correlation of tumor-infiltrating lymphocytes using deep learning on pathology images. *Cell reports*, 23(1):181–193, 2018.

Hidetoshi Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of statistical planning and inference*, 90(2):227–244, 2000.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Gregor Urban, Krzysztof J Geras, Samira Ebrahimi Kahou, Ozlem Aslan, Shengjie Wang, Rich Caruana, Abdelrahman Mohamed, Matthai Philipose, and Matt Richardson. Do deep convolutional nets really need to be deep and convolutional? *arXiv preprint arXiv:1603.05691*, 2016.

Xuezhi Wang and Jeff Schneider. Flexible transfer learning under support and model shift. In *Advances in Neural Information Processing Systems*, pp. 1898–1906, 2014.

Yuxin Wu and Kaiming He. Group normalization. *arXiv preprint arXiv:1803.08494*, 2018.

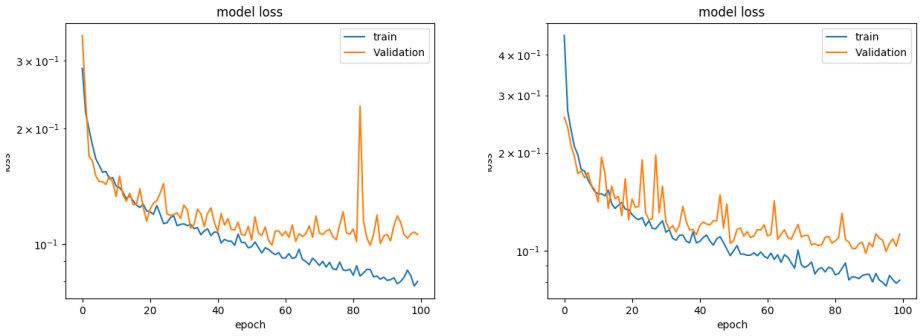
Yan Xu, Zhipeng Jia, Yuqing Ai, Fang Zhang, Maode Lai, I Eric, and Chao Chang. Deep convolutional activation features for large scale brain tumor histopathology image classification and segmentation. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 947–951. IEEE, 2015.

Xiao Xiang Zhu, Devis Tuia, Lichao Mou, Gui-Song Xia, Liangpei Zhang, Feng Xu, and Friedrich Fraundorfer. Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, 5(4):8–36, 2017.

A Appendix

A.1 Implementation Details

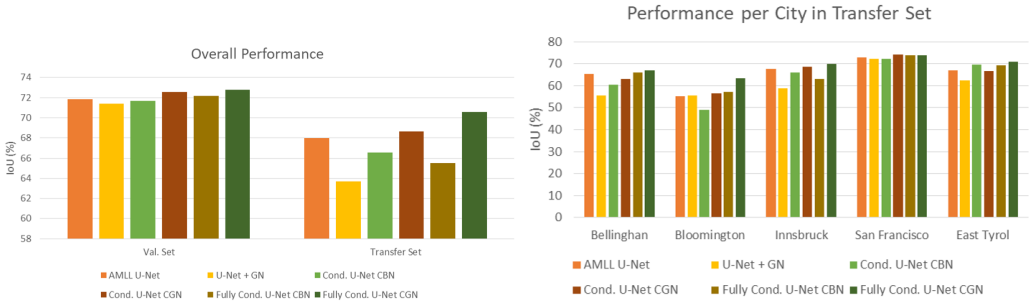
Training Details We trained all networks in this work using 572x572 randomly sampled patches from all training image tiles. We used the Adam optimizer with a batch size of 12. All networks were trained from scratch with a learning rate of 0.001. We keep the same learning rate for the first 60 epochs and decay the rate to 0.0001 over the next 40 epochs. Every network was trained for 100 epochs. In every epoch 8,000 patches are seen. Binary Cross-Entropy was used as the segmentation loss function. For the conditioning part of the conditional network we used the latitude and longitude of every patch center pixel as t. Latitude and longitude were standardised to be from -1 to +1. The dimension of z is 4374 (9x9x54). z is the input of the MLP predicting for γ and β. We used 1024 hidden units in the MLP.



(a) Train loss AMLL U-Net (b) Train loss Cond. U-Net

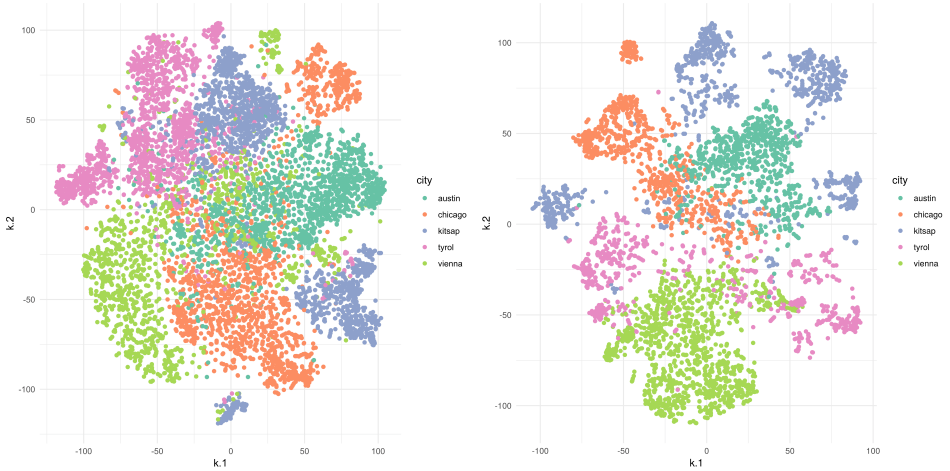
Figure 6: (a) Training loss AMLL U-Net (b)Training loss AMLL U-Net Cond. U-Net

Figure 6 shows the learning curves of the baseline U-Net and the fully conditional U-Net (CGN). The AMLL U-Net seems more prone to overfitting than our conditional U-Net.



(a) Histogram of overall performance (b) Histogram of per city performance

Figure 7: (a) Histogram of overall performance for both validation and transfer set (b) Histogram of per city performance



(a) t-SNE AMLL U-Net train set (b) t-SNE Cond. U-Net train set

Figure 8: t-SNE based on activations obtained using image tiles from the cities in the training set. (a) AMLL U-Net train set (b) Cond. U-Net CGN train set

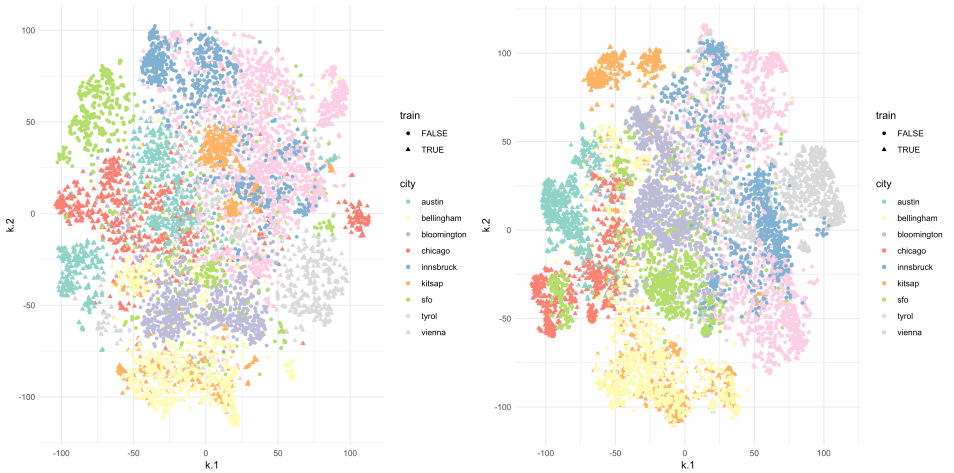
A.2 Detailed Performance

Figure 7 shows the overall and per-city performance of the models. The fully-conditional U-Net variant using CGN consistently outperforms the other models on every city in the transfer set. The CGN variant where only the encoder is conditioned has a stronger overall performance and generalizes significantly better than the conditional U-Net variants using BN. This is consistent with the observation of Wu & He (2018) that (regular) GN outperforms BN in segmentation tasks.

A.3 Network Feature Activations

Figures 8 and 9 are the t-SNE figures obtained from inspecting the activations of the bottom of the U in the U-Net for both the AMLL U-Net and Conditional U-Net CGN. Figure 8 shows the t-SNE associated to sample images from the same cities in the training set. Figure 8 shows the t-SNE associated to sample images from both training and transfer set. We refer to these figures while discussing hypothesis 2 in the Experiments section.

In light of the differences in scale and sparsity of feature activations in the conditional and AMLL U-Nets, we zoom into individual features of interest in Figure 10. It is not surprising that the overall y-axis scale differs between the two networks – this is clear by inspecting the legend of Figure 4. However, we observe that there are more instances of high variation in activation across patches in the conditional U-Net, compared to the AMLL U-Net, among both low and high variance features. Indeed, in the conditional U-Net, there seem to be features that, while typically active, are often



(a) t-SNE AMLL U-Net train and transfer sets (b) t-SNE Cond. U-Net train and transfer sets

Figure 9: t-SNE based on activations obtained using image tiles from cities in both training and transfer sets. (a) AMLL U-Net train and transfer sets (b) Cond. U-Net CGN train and transfer sets

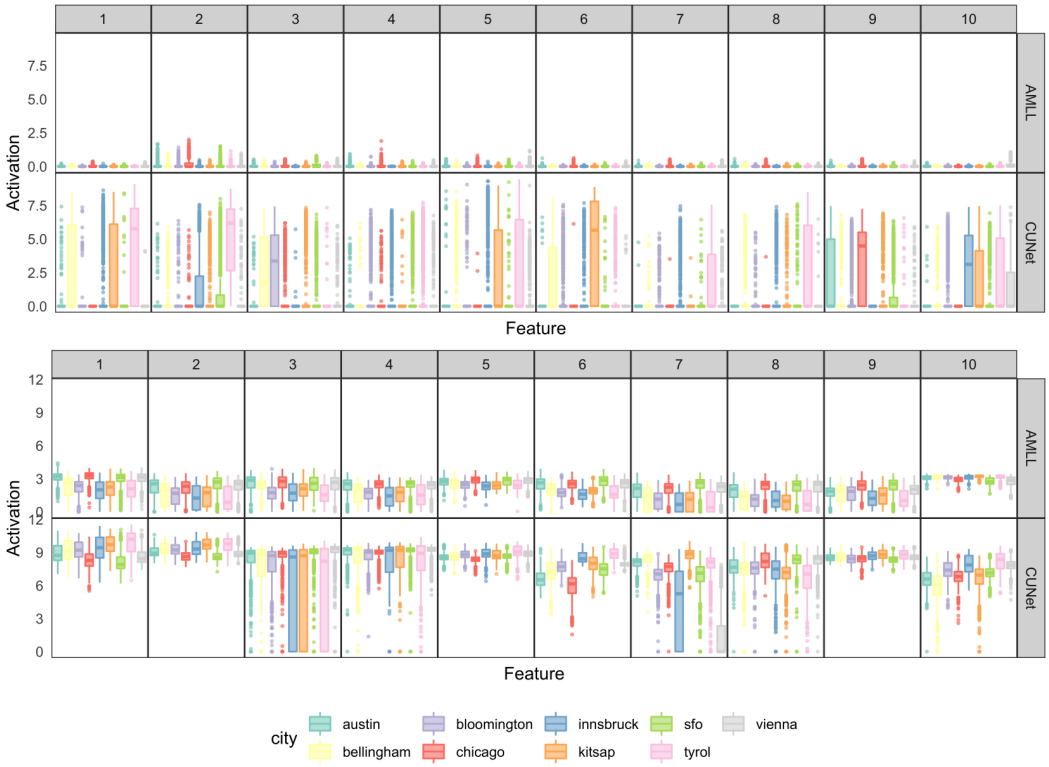


Figure 10: The (top) least and (bottom) most variable features, according to interquartile range, in the AMLL and conditional U-Nets, after filtering to those features that are active in at least 10% of patches. Columns 1-10 give the 10 most and least variable features, for least and most variable features, respectively. The y-axis is the activation for each feature. Patches are split into individual cities, and a boxplot of each city's activation values is given.

exactly or near zero, and similarly, features that are typically inactive, but occasionally spike. This type of variation is even observed within individual cities.