

Multi-Task Classification of Sewer Pipe Defects and Properties using a Cross-Task Graph Neural Network Decoder

Supplementary Materials

Joakim Bruslund Haurum¹

Meysam Madadi²

Sergio Escalera^{1,2,3}

Thomas B. Moeslund¹

¹ Visual Analysis and Perception (VAP) Laboratory, Aalborg University, Denmark

² Computer Vision Center, Autonomous University of Barcelona, Spain

³ Dept. of Mathematics and Informatics, Universitat de Barcelona, Spain

joha@create.aau.dk, mmadadi@cvc.uab.es, sergio@maia.ub.es, tbm@create.aau.dk

A. Content

In these supplementary materials we describe the hyperparameter search, more in-depth results for the optimization-based multi-task learning (MTL) methods as well as the ablation studies. We also show examples of the different task classes, and show examples of success and failure cases for the CT-GNN. Specifically, the following will be described:

- Example images of the different task classes (Section B).
- Hyperparameter search (Section C).
- In-depth optimization-based MTL results (Section D).
- In-depth results for the λ_{defect} ablation study (Section E).
- In-depth results for the MTAN and CT-GNN ablation study (Section F).
- Examples of how the CT-GNN succeeds and fails (Section G).

B. Sewer-ML Task Class Examples

For the sake of clarity we show examples of each class in the water level, pipe shape and pipe material tasks, see Figure 10-12. For examples of the pipe defect classes we refer to the supplementary materials of the Sewer-ML paper [1].

C. Hyperparameter Search

In the hyperparameter search for the CT-GNN decoder we investigated the effect when varying the design of the bottleneck layer and the CT-GNN. The investigated parameters and their search space is presented in Table 2. It should be noted that the amount of attention heads, H , and the re-weighting parameter, p , were only utilized for the GAT

Table 1: Initial Hyperparameter Values. The investigated hyperparameters are set to the following starting values, and after each step of the sequential search the corresponding hyperparameter is updated. It should be noted that τ used in the GAT GNN was set to 0.05. This was done to reduce the amount of noisy graph edges in the Sewer-ML dataset, caused by the large class imbalance in some tasks.

Hyperparameter	GCN	GAT
L	2	2
d_{EMB}	256	256
d_{BTL}	32	32
H	-	8
τ	0.05	0.05
p	0.2	-

[6] and GCN [3] GNNs, respectively. Due to the amount of hyperparameters and the size of the value ranges, we decided to employ a sequential hyperparameter search design. The search was initialized with the hyperparameters stated in Table 1. All tests were performed with $\lambda_{\text{defect}} = 0.50$ to ensure a fair weighting of the tasks, while prioritizing the defect task.

At each step of the search the best performing hyperparameter was kept and used for all future steps of the search. The order of the sequential search was realized as follows:

1. Grid search across the number of GNN layers, L , and the number of GNN channels, d_{EMB} .
2. Search over the number of channels in the bottleneck layer, d_{BTL} .
3. Search over the number of attention heads, H . **Only performed for GAT.**
4. Search over the adjacency matrix threshold, τ .

Table 2: **Investigated Hyperparameters.** The hyperparameters of the CT-GNN and the Bottleneck layer were investigated. For each hyperparameter we have denoted the values investigated.

Hyperparameter	Range
L	[1, 2, 3]
d_{EMB}	[128, 256, 512]
d_{BTL}	[16, 32, 64, 128]
H	[1, 2, 4, 8, 16]
τ	[0.00, 0.05, 0.15, 0.25, 0.35, 0.45, 0.55, 0.65, 0.75, 0.85, 0.95]
p	[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9]

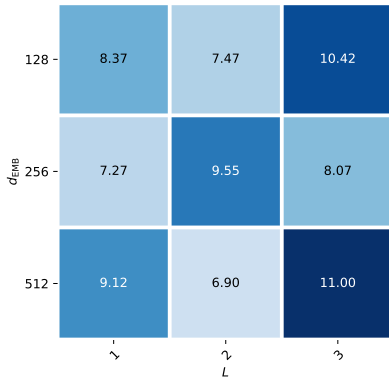


Figure 1: Grid search over L and d_{EMB} for CT-GCN.

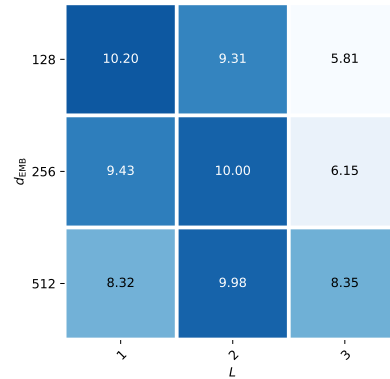


Figure 2: Grid search over L and d_{EMB} for CT-GAT.

5. Search over the adjacency matrix neighbor node reweighting parameter, p . **Only performed for GCN.**

The results of the sequential hyperparameter search on the Sewer-ML dataset are shown in Figures 1-6. From these results we can conclude that the performance when using the GAT leads to more stable performances as the Δ_{MTL} in general does not vary as wildly as when using the GCN. However, when using the GCN we achieve in general higher Δ_{MTL} . We can also observe that the adjacency matrix threshold τ has a large effect on the performance. Specifically, it is observable that using a low τ of 0.05 leads to good performance, which is only matched when τ is set to 0.65 and above for the GAT and 0.35 and above for the GCN. Lastly, we observe that an increased neighbor node reweighting parameter p leads to degraded performance, indicating that the center-node information is crucial. The conditional probability matrix, the binary matrices with τ set to 0.05 and 0.65, as well as the reweighted adjacency matrix with $\tau = 0.05$ and $p = 0.2$ are shown in Figure 7a-7d.

D. Optimization-Based MTL - In-Depth Results

We present the full results for the optimization-based method Dynamic Weight Averaging (DWA) [5] and Uncer-

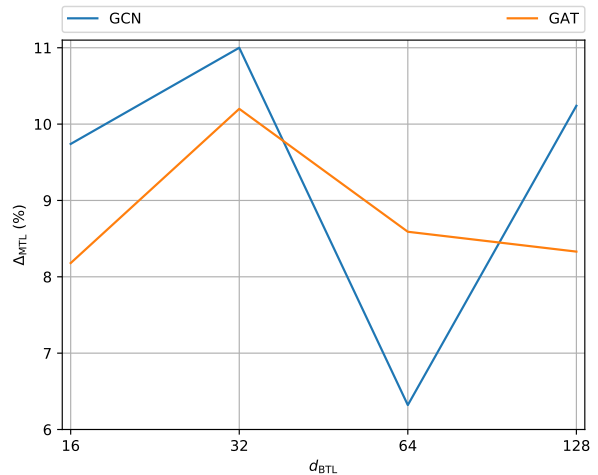


Figure 3: Search over d_{BTL} .

tainty estimation (Uncrt.) [2, 4], see Table 3. The DWA task weighting method is initialized with $\lambda_{\text{defect}} = 0.90$, while Uncrt. is initialized with unit variances for each task. From the results we observe that the DWA method performs worse than the STL networks on nearly every task. The Uncrt. method improves the shape and material MF1 compared to

Table 3: **Effect of optimization-based methods.** In-depth results for two optimization-based methods, DWA [5] and the uncertainty (Uncrt.) based method [2, 4]. TW indicates the task weighting method used and #P indicates the number of parameters in millions. The best performance in each column is denoted in **bold**.

	Model			Overall	Defect		Water		Shape		Material	
	Encoder	TW	#P	Δ_{MTL}	F2 _{CIW}	F1 _{Normal}	MF1	mF1	MF1	mF1	MF1	mF1
Val.	STL	-	94.0	+0.00	58.42	92.42	69.11	79.71	46.55	98.06	65.99	96.71
	R50-MTL	DWA	23.5	-15.70	34.22	86.57	53.43	70.83	37.68	98.18	53.50	90.79
	R50-MTL	Uncrt.	23.5	-4.07	24.80	86.80	62.00	75.31	67.30	99.19	67.46	95.66
Test	STL	-	94.0	+0.00	57.48	92.16	69.87	80.09	56.15	97.59	69.02	96.67
	R50-MTL	DWA	23.5	-11.57	34.84	86.20	54.30	71.03	59.27	97.81	60.39	90.49
	R50-MTL	Uncrt.	23.5	-3.78	26.30	86.48	63.01	76.15	79.69	98.99	70.84	95.59

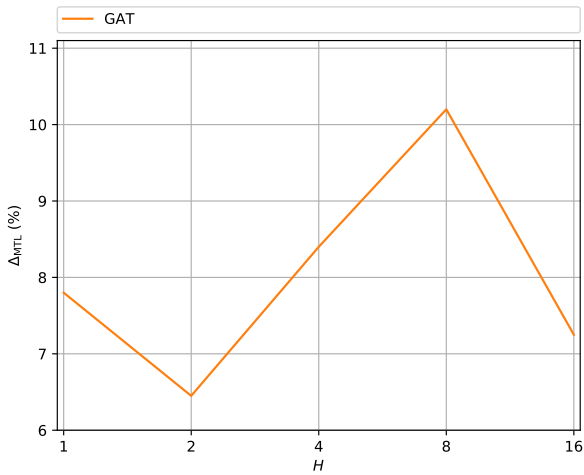


Figure 4: Search over H .

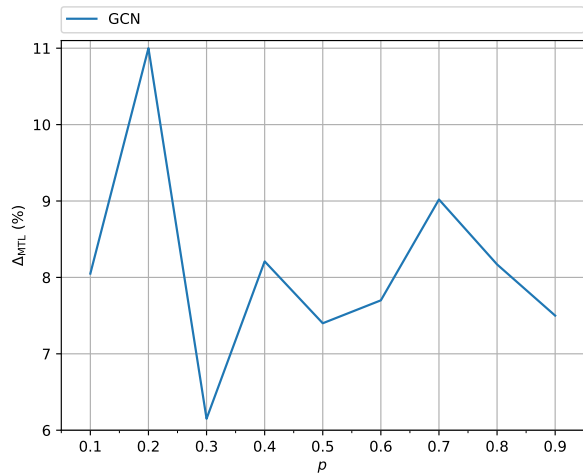


Figure 6: Search over p .

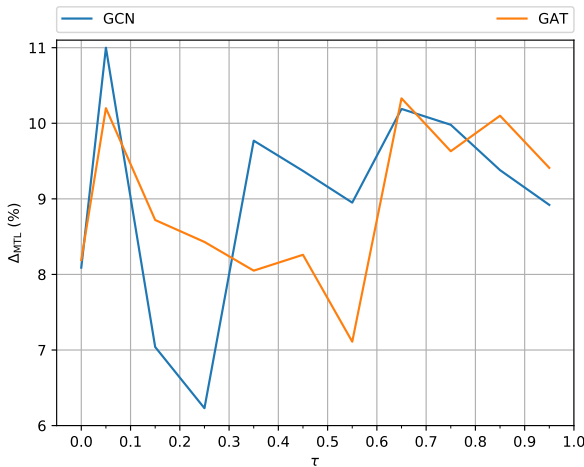


Figure 5: Search over τ .

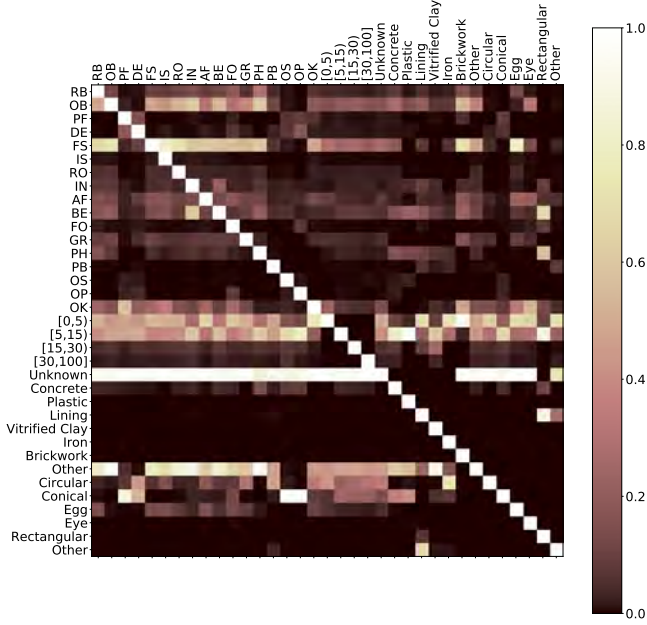
the STL networks, but suffers from poor defect classification rate.

E. Effect of λ_{defect} - In-Depth Results

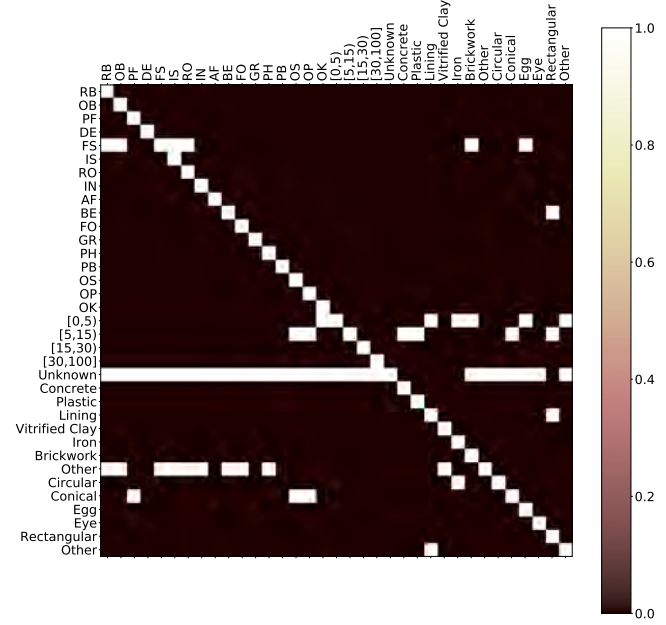
We show the in-depth results for each tested setting of λ_{defect} on the validation split for the R50-MTL baseline as well as CT-GNNs, see Table 4. We observe that a larger λ_{defect} leads to a higher Δ_{MTL} due to a higher F2_{CIW}. However, it also leads to a lower material MF1 score, as we observe that the material MF1 score peaks at 90.5% for the CT-GNNs when $\lambda_{\text{defect}} = 0.50$, and decreases to 82-86% when $\lambda_{\text{defect}} = 0.90$.

F. Combining the MTAN Encoder and CT-GNN Decoder - In-Depth Results

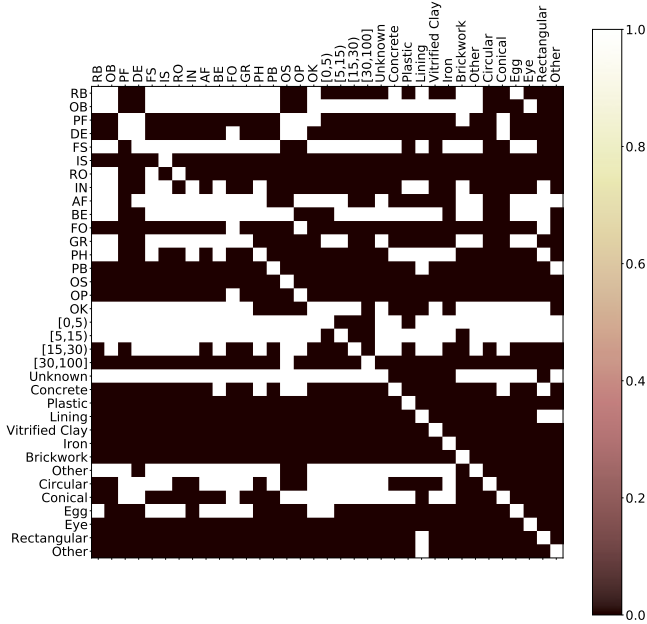
We present the in-depth results of the ablation studies investigating the combination of MTAN encoder and the CT-GNN Decoder, see Table 5. The methods were only



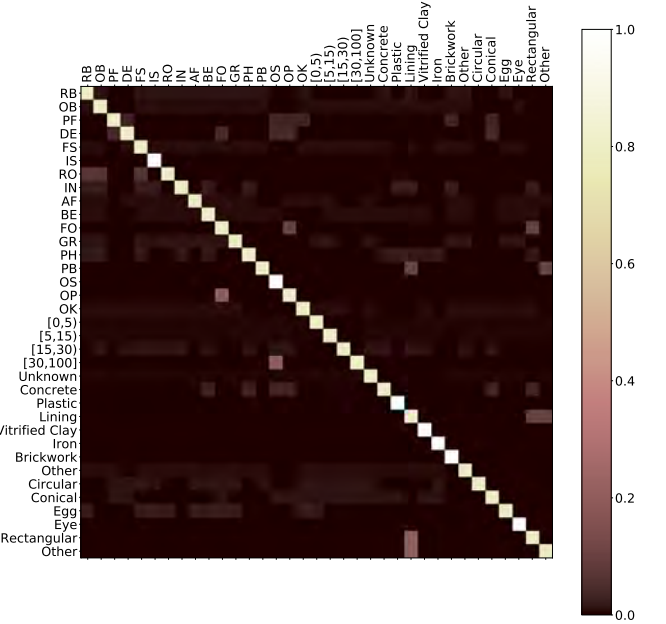
(a) The conditional probability matrix based on the training labels.



(b) The re-weighted adjacency matrix obtained when $\tau = 0.65$.



(c) The re-weighted adjacency matrix obtained when $\tau = 0.05$.



(d) The re-weighted adjacency matrix when $\tau = 0.05$ and $p = 0.2$.

Figure 7: **Adjacency matrix construction.** We show the conditional probability matrix across task classes, as well as the constructed binary and reweighted adjacency matrices.

evaluated on the validation split. From the results we see that the Δ_{MTL} is increased by introducing the CT-GNN, and that the combination with the CT-GCN outperforms using a hard-shared encoder. We observe that the noticeable difference is in the defect classification task where the performance is increased by 0.6-0.7 percentage points on the $F2_{\text{CIW}}$ metric.

G. CT-GNN Success and Failure Cases

We show several cases where the CT-GNN decoder correctly classifies all tasks, shown in Figure 8, as well as cases where some or all tasks are misclassified, shown in Figure 9.

We observe that the the CT-GNN performs well when several defects occur at the same time at different distances

Table 4: **Effect of λ_{defect} .** We compare the performance of the R50-MTL baseline and CT-GNN heads when training with different λ_{defect} values. Evaluated on the validation split. The best performance in each column is denoted in **bold** per method.

Model		Overall	Defect		Water		Shape		Material	
Model	λ_{defect}	Δ_{MTL}	$F2_{\text{CIW}}$	$F1_{\text{Normal}}$	MF1	mF1	MF1	mF1	MF1	mF1
R50-MTL	0.25	+5.45	32.86	88.40	69.42	79.85	74.72	99.21	84.64	97.83
	0.33	+6.22	39.85	89.08	69.18	79.89	70.29	99.31	86.61	97.96
	0.50	+6.91	40.78	89.31	69.35	79.90	71.58	99.25	87.21	97.75
	0.67	+11.11	52.53	90.69	70.19	80.22	75.74	99.38	87.83	98.11
	0.75	+9.99	56.31	91.41	70.15	80.42	69.91	99.40	85.13	98.30
	0.90	+10.36	59.73	91.87	70.51	80.47	71.64	99.34	80.28	98.09
	0.95	+10.40	60.34	91.85	69.35	80.02	71.85	99.19	81.26	97.82
CT-GCN	0.25	+5.79	39.44	88.77	69.76	79.63	73.67	99.27	80.06	97.79
	0.33	+7.45	42.56	89.12	69.36	79.82	72.20	99.20	87.40	97.81
	0.50	+11.00	50.35	90.01	70.04	79.98	75.80	99.44	90.54	97.96
	0.67	+10.20	54.67	90.64	69.78	79.92	72.94	99.35	85.31	98.06
	0.75	+10.75	57.71	91.11	70.48	80.21	70.95	99.37	86.27	98.21
	0.90	+12.39	61.35	91.84	70.57	80.47	76.17	99.33	82.63	98.18
	0.95	+9.05	62.10	92.01	69.95	80.04	67.36	99.11	77.83	97.89
CT-GAT	0.25	+7.69	37.02	88.69	70.06	80.18	75.47	99.45	89.40	97.89
	0.33	+5.70	42.17	89.09	69.54	79.96	71.72	99.37	78.80	97.95
	0.50	+10.33	49.96	89.98	69.69	79.90	73.90	99.41	90.52	98.06
	0.67	+10.20	55.26	90.69	69.80	80.38	72.41	99.40	84.90	98.12
	0.75	+12.10	58.37	91.45	70.46	80.43	76.82	99.46	83.75	98.35
	0.90	+12.81	61.70	91.94	70.57	80.43	74.53	99.40	86.63	98.24
	0.95	+10.65	60.95	92.03	69.01	79.59	70.75	99.18	83.99	97.84

Table 5: **Effect of encoder.** We compare the effect of training CT-GNN using GCN and GAT with the MTAN encoder, and with fixed task weights. #P indicates the number of parameters in millions. Evaluated on the validation split. The best performance in each column is denoted in **bold**.

Model		Overall	Defect		Water		Shape		Material		
Encoder	CT-GNN	#P	Δ_{MTL}	$F2_{\text{CIW}}$	$F1_{\text{Normal}}$	MF1	mF1	MF1	mF1	MF1	mF1
MTAN	X	48.2	+10.40	61.21	92.10	70.06	80.59	68.34	99.40	83.48	98.25
MTAN	GCN	49.9	+12.72	61.86	91.99	71.39	80.53	75.42	99.46	83.77	98.25
MTAN	GAT	48.6	+11.48	61.92	92.03	70.95	80.50	71.17	99.39	83.65	98.29

to the camera (see top left example), as well as subtle defects such as the distortion in the bottom middle example and crack in the bottom left example. Similarly, this can be observed in the top right example where the high water level is detected even though it is partially occluded and unlit. Lastly it can correctly handle rare classes such as the iron material in the bottom right example.

In Figure 9 we observe that the the CT-GNN misclassify irregularities in the pipe geometry as displaced pipes (FS) or construction changes (OK), as seen in the top right and top middle examples. In both cases the predictions is understandable as the internal reparation is shifted (top left) and the

camera is placed right before a well (top middle). In the top right case the deformation is observed as a surface damage, which is understandable due to the folds of the deformation. For the cases where all classifications are incorrect, we see that the CT-GNN decoder misclassifies several tasks due to limited context introduced by the camera perspective.

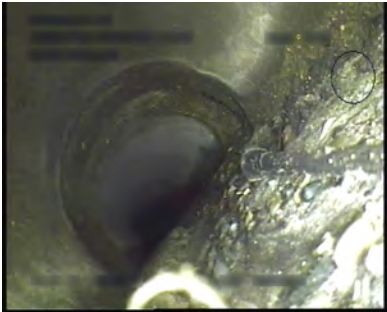





								
Task	Ground Truth	CT-GNN	Task	Ground Truth	CT-GNN	Task	Ground Truth	CT-GNN
Defect	RB,OB,FS,AF	RB,OB,FS,AF	Defect	FS,AF	FS,AF	Defect	FS,PH	FS,PH
Water	[0%, 5%)	[0%, 5%)	Water	[5%, 15%)	[5%, 15%)	Water	[30%, 100%]	[30%, 100%]
Shape	Circular	Circular	Shape	Circular	Circular	Shape	Circular	Circular
Material	Concrete	Concrete	Material	Concrete	Concrete	Material	Concrete	Concrete
								
Task	Ground Truth	CT-GNN	Task	Ground Truth	CT-GNN	Task	Ground Truth	CT-GNN
Defect	RB,PB	RB,PB	Defect	DE	DE	Defect	OB,OK	OB,OK
Water	[5%, 15%)	[5%, 15%)	Water	[5%, 15%)	[5%, 15%)	Water	[0%, 5%)	[0%, 5%)
Shape	Circular	Circular	Shape	Circular	Circular	Shape	Circular	Circular
Material	Plastic	Plastic	Material	Lining	Lining	Material	Iron	Iron

Figure 8: **Examples of correct classifications with the CT-GNN.** Example cases where the CT-GNN correctly classifies all four tasks.

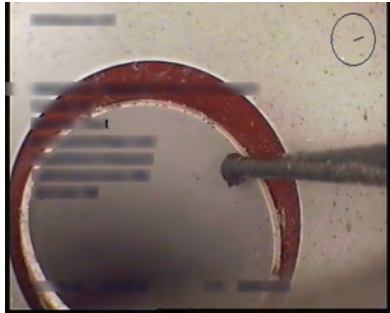

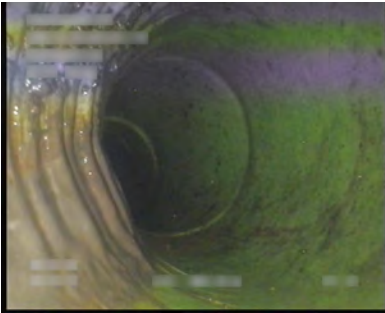
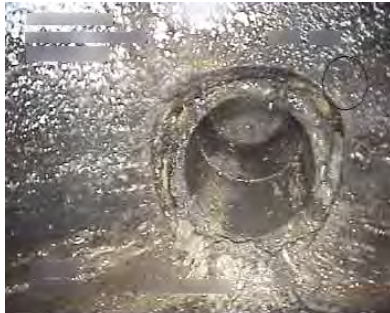
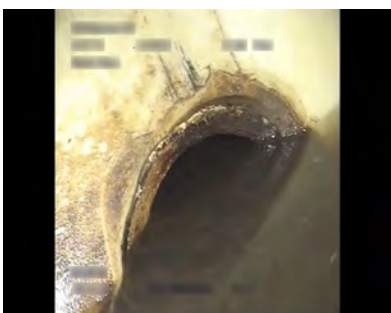

								
Task	Ground Truth	CT-GNN	Task	Ground Truth	CT-GNN	Task	Ground Truth	CT-GNN
Defect	OK	OK,FS	Defect	BE	BE,OK	Defect	DE,OK	OB,OK
Water	[0%, 5%)	[0%, 5%)	Water	[0%, 5%)	[5%, 15%)	Water	[0%, 5%)	[0%, 5%)
Shape	Circular	Circular	Shape	Circular	Circular	Shape	Circular	Circular
Material	Plastic	Plastic	Material	Plastic	Plastic	Material	Lining	Lining
								
Task	Ground Truth	CT-GNN	Task	Ground Truth	CT-GNN	Task	Ground Truth	CT-GNN
Defect	PF,OS	OB,FS,PH	Defect	OS	None	Defect	OK	None
Water	[5%, 15%)	[30%, 100%]	Water	[15%, 30%)	[30%, 100%]	Water	[5%, 15%)	[0%, 5%)
Shape	Conical	Circular	Shape	Circular	Conical	Shape	Circular	Conical
Material	Lining	Concrete	Material	Plastic	Lining	Material	Plastic	Lining

Figure 9: **Examples of incorrect classifications with the CT-GNN.** Example cases where the CT-GNN incorrectly classifies some or all four tasks. Incorrect classifications are denoted in **red**.

Figure 10: **Water level class examples.** Example images of the four considered water level classes.

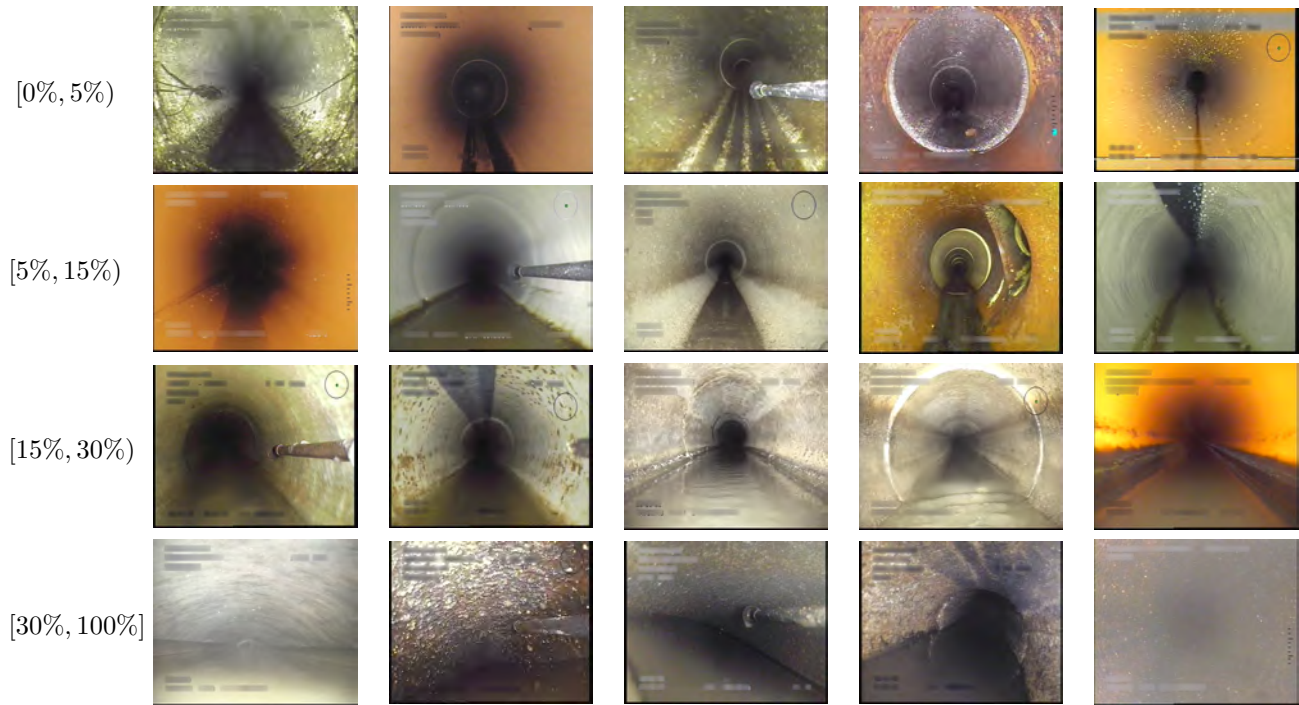
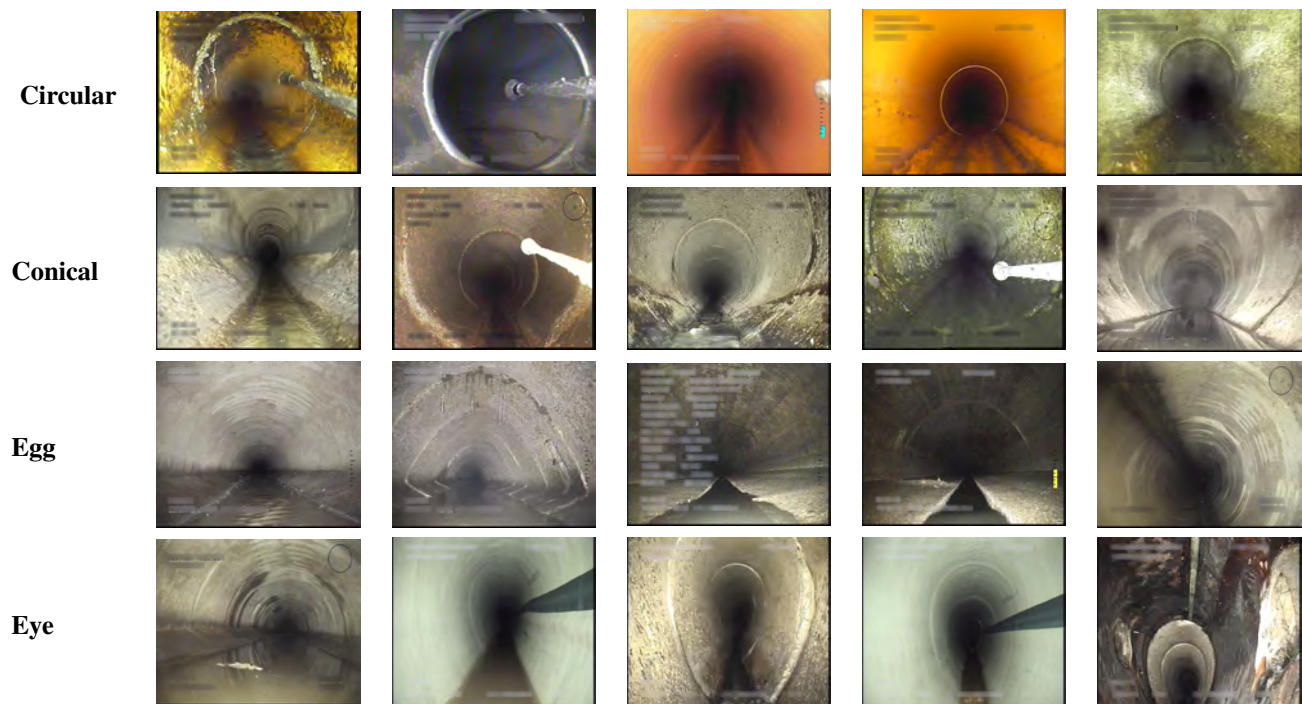


Figure 11: **Pipe shape class examples.** Example images of the six considered pipe shape classes.

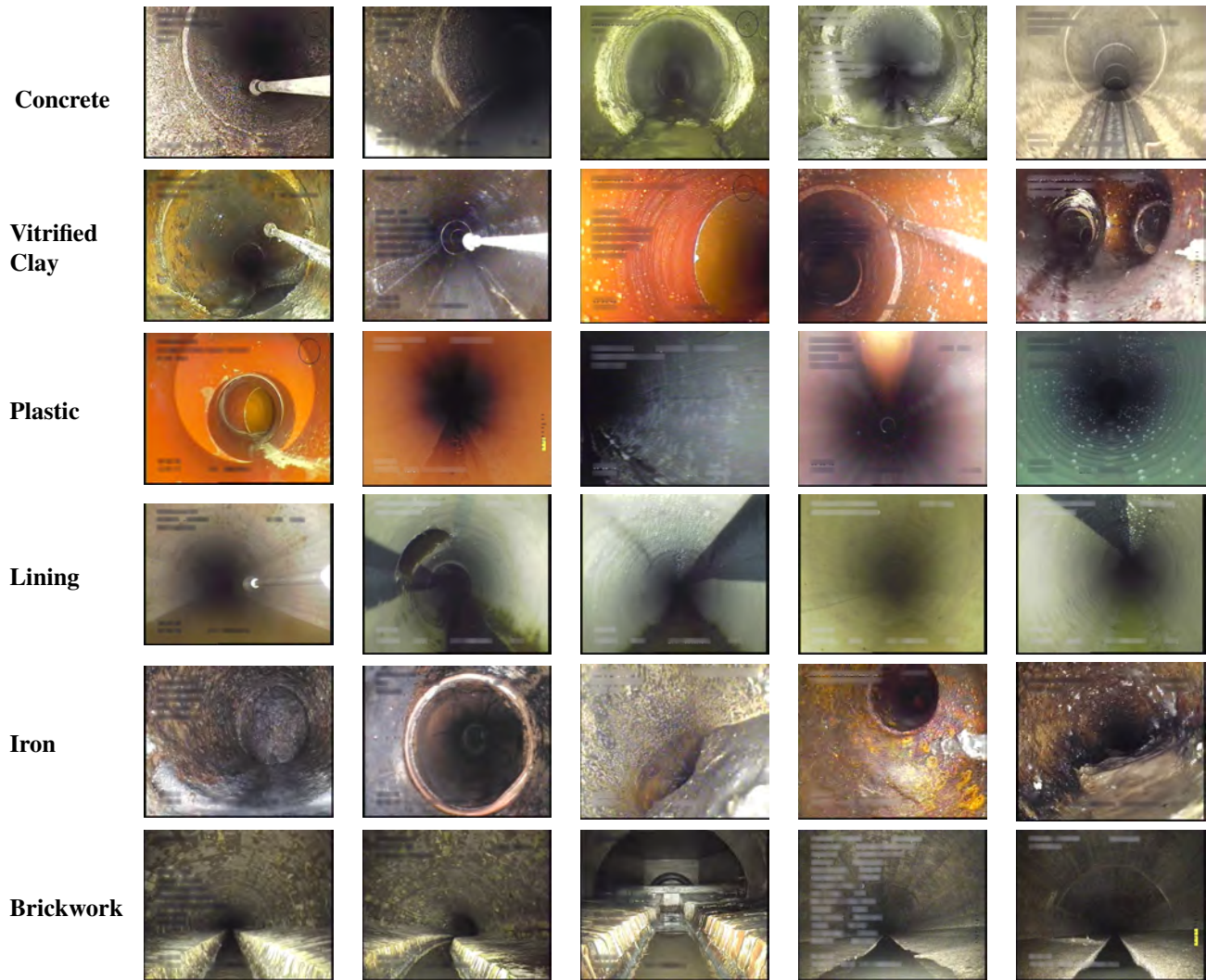


Continued on next page

Figure 11: Continued from previous page



Figure 12: **Pipe material class examples.** Example images of the eight considered pipe material classes.



Continued on next page

Figure 12: Continued from previous page

Unknown



Other



References

- [1] Joakim Bruslund Haurum and Thomas B. Moeslund. Sewerml: A multi-label sewer defect classification dataset and benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.
- [2] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [3] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.
- [4] Lukas Liebel and Marco Körner. Auxiliary tasks in multi-task learning, 2018. arxiv: 1805.06334.
- [5] Shikun Liu, Edward Johns, and Andrew J. Davison. End-to-end multi-task learning with attention. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1871–1880, 2019.
- [6] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *International Conference on Learning Representations*, 2018.