Neural Inverse Knitting: From Images to Manufacturing Instructions

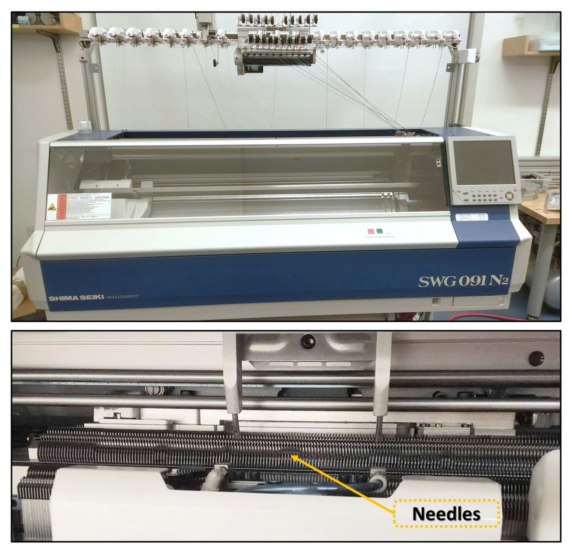
Alexandre Kaspar*, Tae-Hyun Oh*, Liane Makatura, Petr Kellnhofer and Wojciech Matusik

Massachusetts Institute of Technology (MIT), Computer Science and Artificial Intelligence Laboratory (CSAIL)

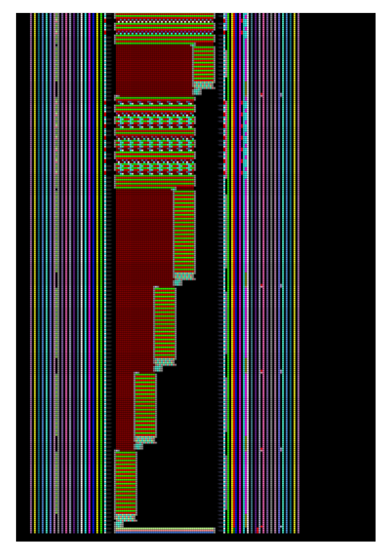
More on deepknitting.csail.mit.edu

Industrial Knitting

- Full customization of whole knitted garments (no sewing needed!)
- Electronic control of every loop of yarn
- BUT complex programming => requires skilled technicians

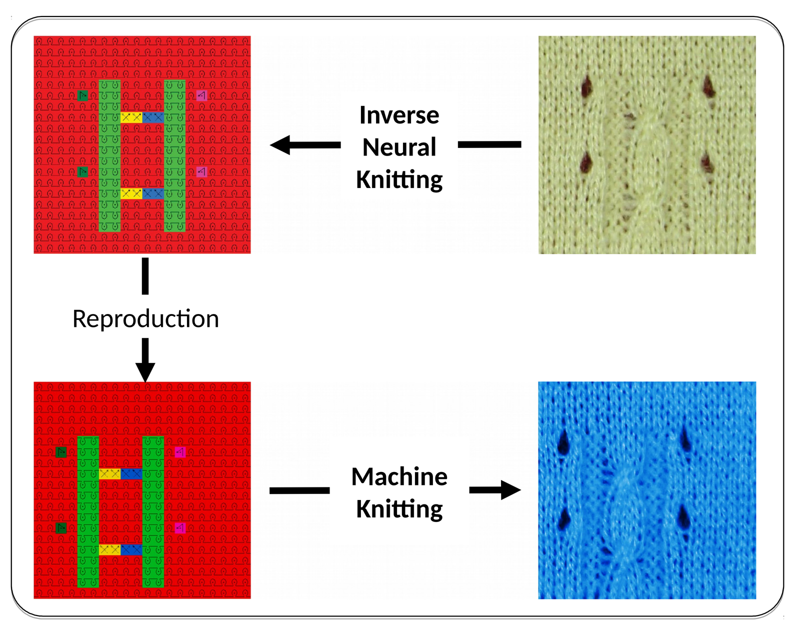


Knitting machine



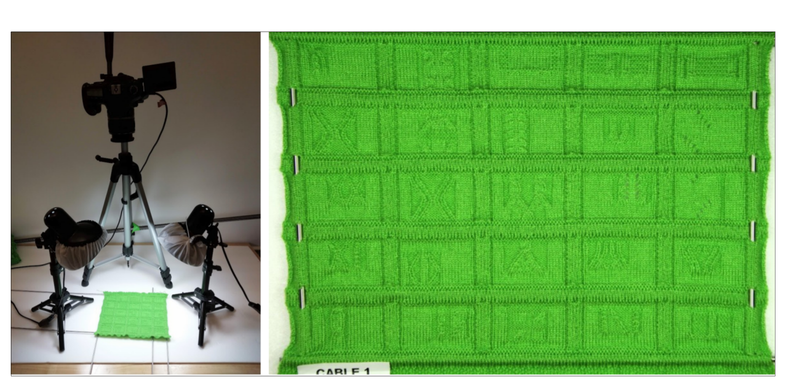
Program map

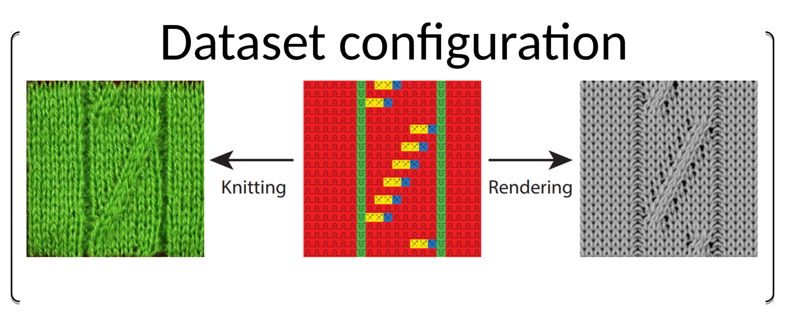
Application Scenario

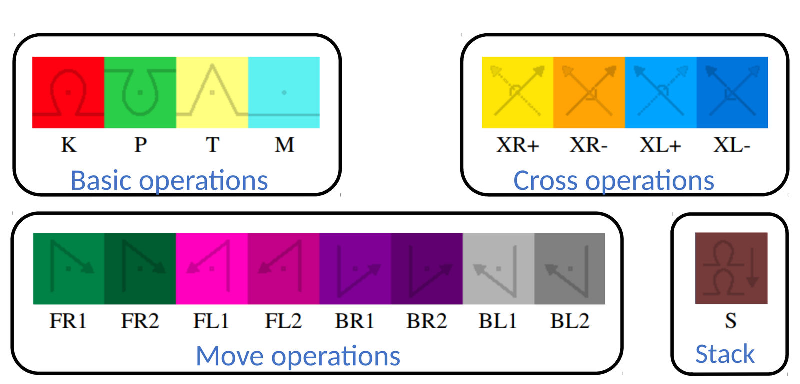


The goal of this work is to automatically **synthesize manufacturing instructions** for a given knitting pattern from an image input.

Knitting Pattern Dataset

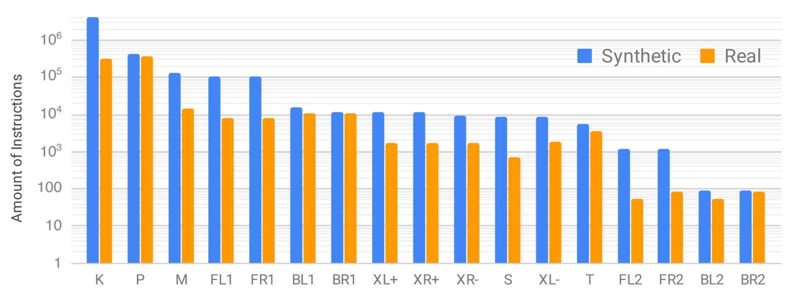






Domain Specific Language (DSL) for regular knitting patterns

- Minimal instruction set
- Unambiguous actions

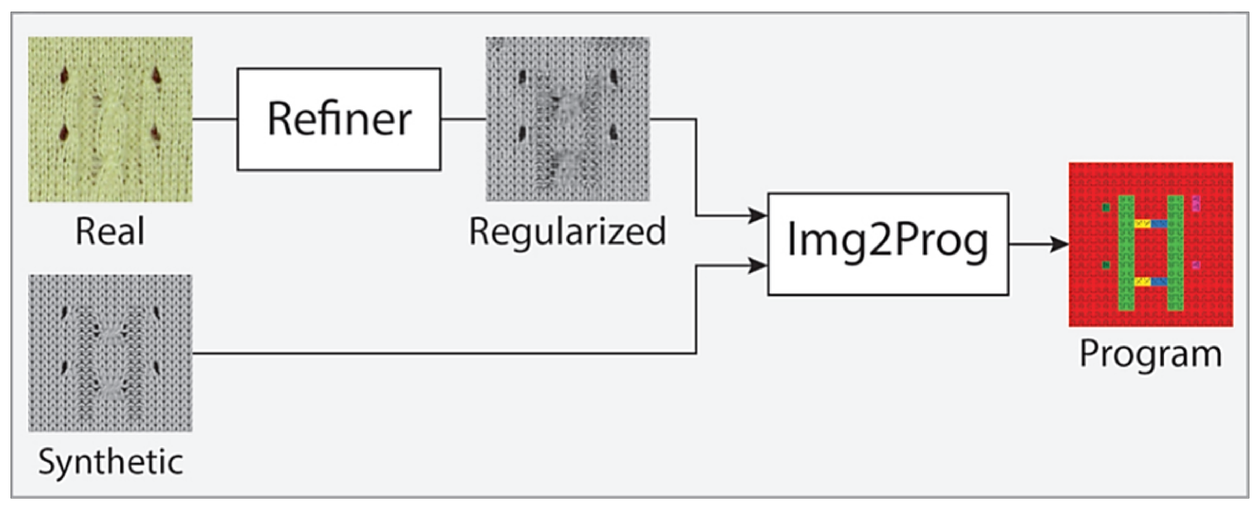


Statistics

- 17 distinct instructions
- 2'088 real samples
- 14'440 synthetic simulations

Method

- Real data is hard to obtain + various valid perturbations
- Training with both real & synthetic
- Key idea: Real to synthetic translation



Syn2Real -vs- Real2Synthetic

- Generalization bound with heterogeneous dataset

Proposition. Generalization Bound with Heterogeneous Data

- $\mathcal{H}$: Hypothesis set
- $\mathcal{D}_S, \hat{S}$: Source distribution, and its sampled set of size βm
- $\mathcal{D}_T, \hat{T}$: Target distribution, and its sampled set of size $(1-\beta)m$
- $y(\cdot)$: True label function
- $\mathcal{L}$: Loss function (holding symmetry and the triangle inequality, e.g., 0-1 loss).

For $\alpha \in [0,1]$, let

$$\hat{h} = \arg\min_h \alpha \mathcal{L}_{\hat{S}}(h,y) + (1-\alpha)\mathcal{L}_{\hat{T}}(h,y), \quad (1)$$

$$h_T^* = \arg\min_h \mathcal{L}_T(h,y). \quad (2)$$

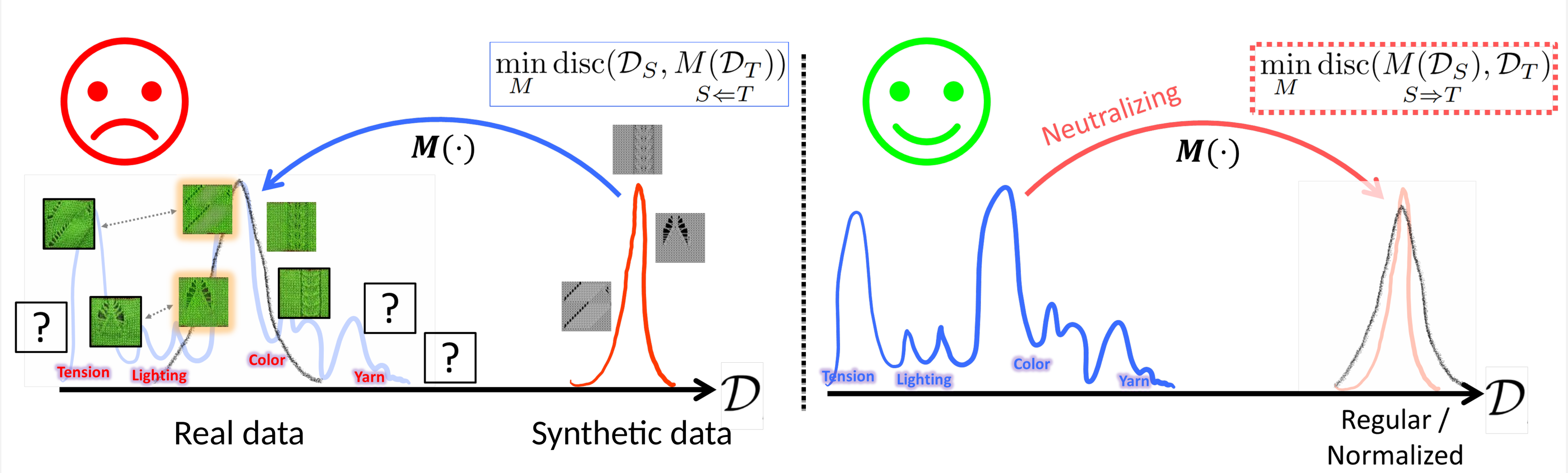
Then, with probability at least $1-\delta \in (0,1)$, we have

$$\tfrac{1}{2}|\mathcal{L}_T(\hat{h},y) - \mathcal{L}_T(h_T^*,y)| \le \alpha\left(\mathrm{disc}_{\mathcal{H}}(\mathcal{D}_S,\mathcal{D}_T) + \lambda\right) + \epsilon, \quad (3)$$

where $\epsilon(m,\alpha,\beta,\delta) = \sqrt{\frac{1}{2m}\left(\frac{\alpha^2}{\beta} + \frac{(1-\alpha)^2}{1-\beta}\right)\log(\frac{2}{\delta})}$, and $\lambda = \min_{h\in\mathcal{H}} \mathcal{L}_S(h,y) + \mathcal{L}_T(h,y)$.

Derived from [Ben-David et al.'10, Mansour et al.'09]

- Focus on discrepancy disc(·,·) between two data distributions
- Introduce a learnable M(·) for optimizing disc(·,·)

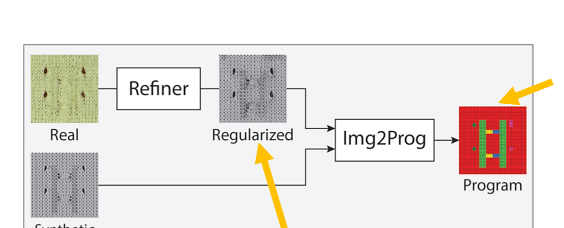


- Adversarial loss emerging

$$\hat{h}, \hat{M} = \arg\min_{h,M} \hat{\mathcal{L}}_\alpha(h,y) + \tau \cdot \mathrm{disc}_{\mathcal{H}}(\hat{\mathcal{D}}_S, M \circ \hat{\mathcal{D}}_R)$$

$$M^* = \arg\min_M \mathrm{disc}_{\mathcal{H}}(\hat{\mathcal{D}}_S,\ M \circ \hat{\mathcal{D}}_R)$$
$$= \arg\min_M \max_{h,h'\in\mathcal{H}} |\mathcal{L}_{\hat{\mathcal{D}}_S}(h,h') - \mathcal{L}_{M\circ\hat{\mathcal{D}}_R}(h,h')|$$

- Actual loss used to train the networks



Results

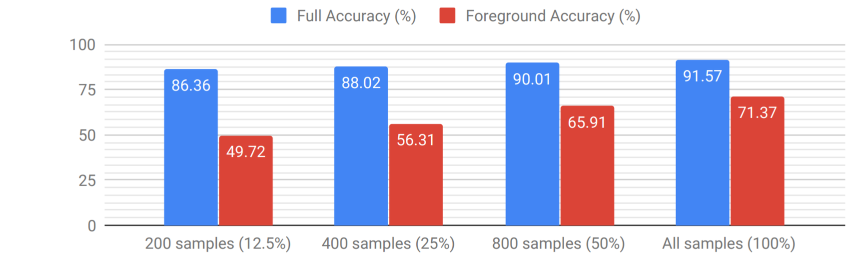
Quantitative comparison

	Method	Accuracy (%) Full	Accuracy (%) FG	Perceptual SSIM	Perceptual PSNR [dB]
(a1)	CycleGAN (Zhu et al., 2017)	46.21	21.58	0.631	15.43
(a2)	Pix2Pix (Isola et al., 2017)	57.11	46.06	0.662	15.94
(a3)	UNet (Ronneberger et al., 2015)	89.46	63.79	0.848	21.79
(a4)	Scene Parsing (Zhou et al., 2018)	87.53	66.38	0.850	21.79
(a5)	S+U (Shrivastava et al., 2017)	**91.85**	**71.47**	**0.872**	**21.93**
(b1)	Img2prog (real only) with CE	91.45	70.73	0.866	21.52
(b2)	Img2prog (real only) with MILCE	**91.94**	**71.61**	**0.875**	**21.68**
(c1)	Refiner + img2prog ($\alpha = 0.1$)	93.62	78.06	0.896	22.90
(c2)	Refiner + img2prog ($\alpha = 0.5$)	93.48	78.47	0.893	23.18
(c3)	Refiner + img2prog ($\alpha = 2/3$)	**94.11**	**81.08**	**0.902**	**23.68**
(c4)	Refiner + img2prog ($\alpha = 0.9$)	91.87	71.44	0.873	21.96
(d1)	Refiner + img2prog++ ($\alpha = 2/3$)	**94.35**	**81.96**	**0.905**	**24.06**

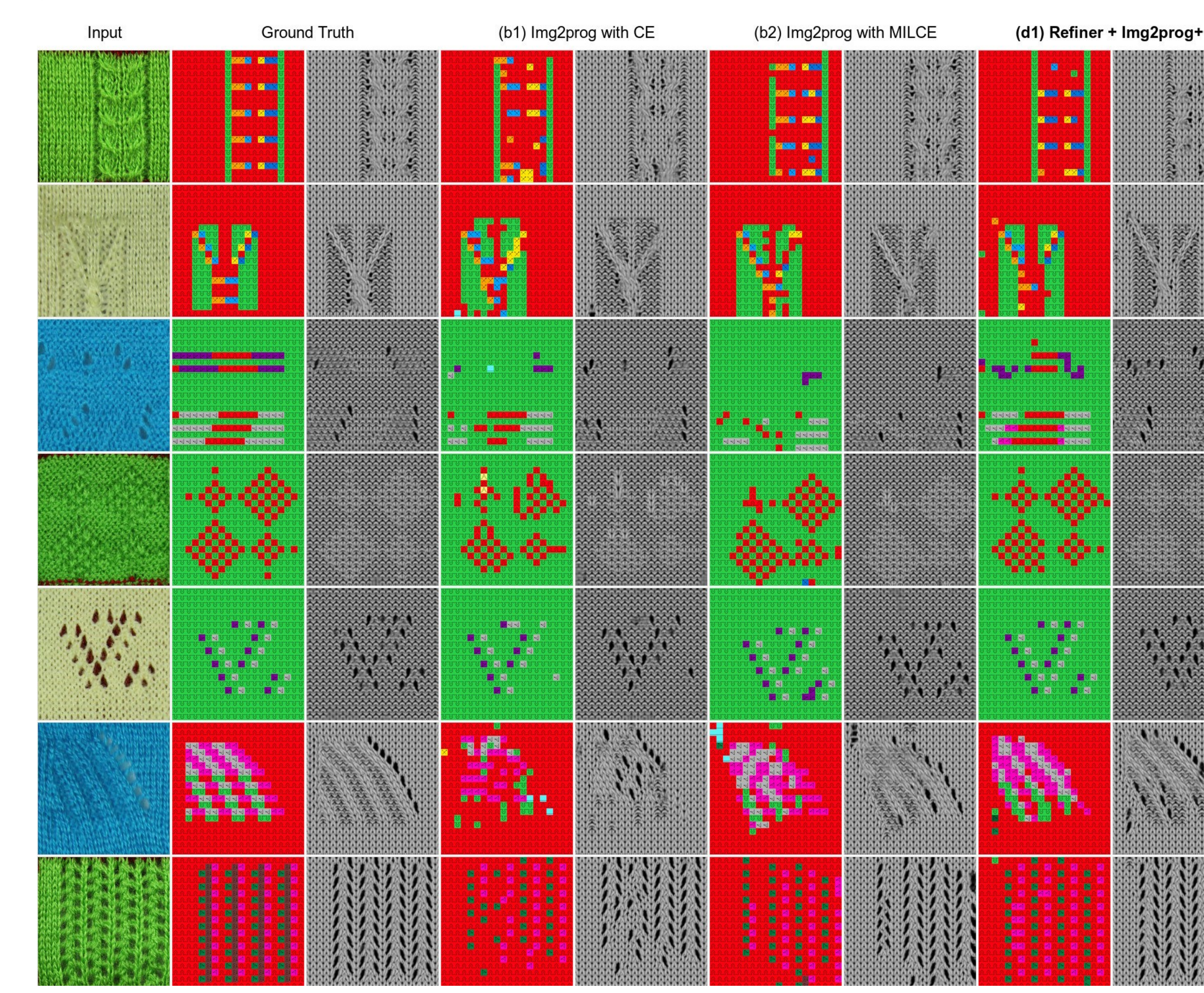
Accuracy breakdown per instruction

Instruction	K	P	T	M	FR1	FR2	FL1	FL2	BR1	BR2	BL1	BL2	XR+	XR-	XL+	XL-	S
Accuracy [%]	96.49	96.58	74.84	71.69	80.22	83.33	76.01	100	71.42	27.27	70.88	27.27	55.21	62.32	62.61	59.28	25.87
Frequency [%]	46.42	45.34	0.50	1.99	1.10	0.01	1.13	0.01	1.08	0.01	1.23	0.01	0.28	0.21	0.26	0.23	0.20

Effects of the number of real samples



Qualitative results



Limitations and Future Work

Limitations

- No hard constraints for manufacturability
- No explicit treatment of stitch scale
- Dataset only uses single type of yarn - acrylic Tamm 2/30 (no fuzzier yarn type, various plies, lace types, etc.)

Future work

- Integrate differentiable renderer during training
- Trade-off between quality and regularity for the synthetic data during training
- Applying to yarn patterns on 3D garments