Learning to Read Braille: Bridging the Tactile Reality Gap with Diffusion Models

Carolina Higuera[1], Byron Boots[1], and Mustafa Mukadam[2]

[1]University of Washington, [2]Meta AI

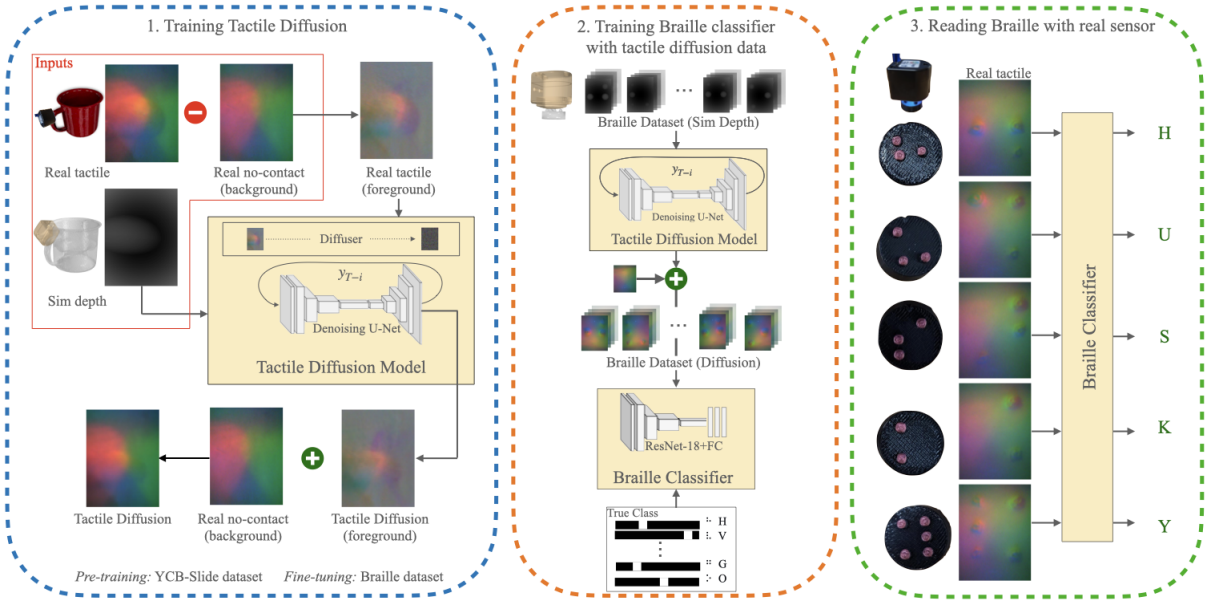

Fig. 1: (Left) Our tactile diffusion model, trained on YCB-Slide and fine-tuned on 20% real braille, learns to generate realistic tactile images from simulated contact depth. (Middle) We train a braille classifier with data generated from tactile diffusion. (Right) On reading real braille with a DIGIT sensor this classifier outperforms classifiers trained with simulation and other approaches.

***Abstract*— Simulating vision-based tactile sensors enables learning models for contact-rich tasks when collecting real-world data at scale can be prohibitive. However, modeling the optical response of the gel deformation as well as incorporating the dynamics of the contact makes sim2real challenging. Prior works have explored data augmentation, fine-tuning, or learning generative models to reduce the sim2real gap. In this work, we present the first method to leverage probabilistic diffusion models for capturing complex illumination changes from gel deformations. Our tactile diffusion model is able to generate realistic tactile images from simulated contact depth bridging the reality gap for vision-based tactile sensing. On real Braille reading task with a DIGIT sensor, a classifier trained with our diffusion model achieves 75.74% accuracy outperforming classifiers trained with simulation and other approaches. Project page: https://github.com/carolinahiguera/Tactile-Diffusion**

I. Tactile Diffusion Model

A. Tactile diffusion pre-training

We use the `YCB-Slide` dataset [1], which consists of sliding contact interactions between a real DIGIT sensor and 10 YCB objects. With the sensor poses and the object meshes, we use TACTO simulator to recreate each trajectory and collect the simulated depth image for each timestep. Our dataset for training the diffusion model consists of 180k aligned pairs of sim depth and real tactile images, split into 80% for training and 20% for testing.

For training our tactile diffusion model, we follow the pipeline shown in Fig. 1 (left). We pre-process the ground-truth data by subtracting from all images a real no-contact image from the sensor. The decoder of the tactile diffusion model uses a conditional U-Net backbone with 2 ResNet blocks for each stage of down-sampling and up-sampling respectively. We allow conditioning on the sim depth image via concatenation, following [2], [3]. We use a linear noise scheduler of $(1e^{-4}, 0.02)$ with $T = 500$ timesteps for training and inference. We use Adam [4] optimizer with a learning rate of $1e^{-4}$. For inference, we start the denoising process from pure Gaussian noise with the sim depth image as input to condition the model. The intuition is to query the model about how the sim depth image will look when using a real sensor. After inference, we post-process the generated foreground by adding the no-contact image back. For training and inference we use a RTX-3080 GPU and the inference time of a batch of 30 images is 23 seconds.

B. Tactile diffusion fine-tuning

We fine-tune our tactile diffusion model with pairs of (sim depth, real) images from braille contact interactions. Our datasets for fine-tuning and testing consist of real tactile

TABLE I: Metrics on braille classification task.

Training data source	% real data fine-tuning	Accuracy %	Precision	Recall
Sim	-	30.23	0.34	0.30
	20	64.99	0.71	0.65
	80	73.11	0.80	0.73
	100	73.95	**0.81**	0.74
Sim + data aug.	-	43.48	0.61	0.43
	100	73.23	0.76	0.73
cGAN	-	31.18	0.40	0.31
Tactile diffusion	-	**75.74**	0.79	**0.76**
Real	-	100.0	1.00	1.00

Training cGAN on 100% real, tactile diffusion on YCB-Slide + 20% real

data collected from a DIGIT sensor when in contact with 27 3D-printed braille characters (letters A-Z and #). We fine-tune the tactile diffusion model previously trained on the `YCB-Slide` with 20% of the fine-tuning dataset. In Fig. 2 we show, for different letters, samples of sim, real, and generated tactile images by tactile diffusion and cGAN as a baseline. Although the use of cGAN to generate tactile images is common in the literature, the models are not open-sourced, thus we are using our implementation of cGAN, conditioning on the same sim depth image.

Qualitatively, tactile diffusion can represent the corresponding gel deformation by rendering the color changes around the object indentations with high detail level. These indentations correspond to the bumps that characterize each braille character. cGAN can do it as well for some characters but exhibits errors rendering the level of gel deformation or skipping its representation. This is congruent with the findings in [5], where the authors explain that GANs can trade off diversity for fidelity, producing high-quality samples, but not covering the whole distribution. In general, we found cGAN more difficult to condition and exhibit less level of texture detail in comparison with the images generated by the diffusion model. Our tactile diffusion model also presents some differences with respect to the real image. However, these differences mostly correspond to misalignment in the location of the bump's indentation.

C. Tactile diffusion for reading braille

We trained this classifier using all current common approaches for tackling sim2real when using vision-based tactile sensing. This consists of training the model using data from sim + data augmentation (lighting randomization), sim + fine-tuning with real data, sim + data augmentation + fine-tuning with real data, cGAN, and tactile diffusion. For the last two approaches, we performed data adaptation on the sim depth images, following the pipeline in Fig. 1 (middle). Table I shows the performance of all these models (accuracy, precision, and recall) when testing the models on our test braille dataset.

Using real data to train the downstream task would be the ideal as we guarantee that both train and test data are under the same distribution. The braille classifier trained directly on real data can perfectly distinguish the letters based on the imprints of the braille characters on the sensor's gel. However, collecting real data is expensive and time-consuming and prohibitive to scale. In practice, we would

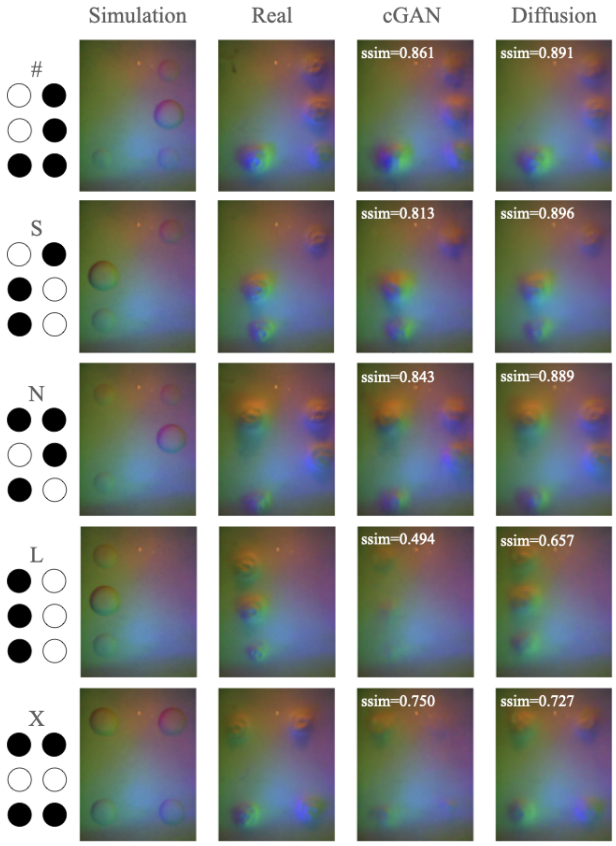

Fig. 2: Example comparisons of sim, real, cGAN, and tactile diffusion images from the braille test dataset. Tactile diffusion consistently generates images with higher SSIM with respect to the real sensor image. Notice that tactile diffusion does not skip the representation of any dots in the braille character.

like to be able to train the downstream task solely in simulation and achieve good performance when deployed in real. Transferring directly the model trained on raw simulation highlights the sim2real gap when working with vision-based tactile data. Data augmentation improves the generalization of the model but not enough to induce the distribution of the real data on the training dataset. Fine-tuning the model on real data definitely helps to improve its performance. We investigate when most performance can be achieved with the least amount of real fine-tuning data. Tactile diffusion is trained on a general dataset of contacts from YCB-Slide and fine-tuned with only 20% of the task relevant braille data. Under these conditions, it achieves a zero-shot accuracy of 75.74% on real tactile data. Fine-tuning the simulation model to achieve similar performance requires collecting 4 times more real tactile data.

Comparing data adaptation techniques, tactile diffusion has significantly better performance than cGAN. This is expected since cGAN sometimes skips the representation of bumps for some braille characters. These anomalies in the tactile image can instead represent a different character leading to missclassification, which hurts the downstream task. Overall, these results highlight the promising future of tactile diffusion to close the sim2real gap for vision-based tactile sensors.

ACKNOWLEDGMENT

The authors thank Sudharshan Suresh, Mike Lambeta, and Roberto Calandra for help with TACTO simulator and DIGIT sensors.

REFERENCES

[1] S. Suresh, Z. Si, S. Anderson, M. Kaess, and M. Mukadam, "Midas-touch: Monte-carlo inference over distributions across sliding touch," *arXiv preprint arXiv:2210.14210*, 2022.

[2] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi, "Palette: Image-to-image diffusion models," in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022, pp. 1–10.

[3] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image super-resolution via iterative refinement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[4] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[5] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.