Appendix A Networks Architectures

A.1 Mesh Network

Our 3D mesh disentangled variational autoencoder (β-VAE) uses a graph Convolutional autoencoder (CoMA) architecture, which leverages Fourier transformations to extract spectral information from the mesh graph. This information is then processed using Chebyshev filters, enabling localized convolution operations (Defferrard et al. 2016). The encoder module consists of four graph convolutional layers, each with filter sizes of [16, 16, 16, 32] and a kernel size of 6. Downsampling operations with a factor of 2 progressively reduce the mesh's spatial dimensions. We use $ReLU$ activation functions. The latent space is parameterized by two parallel linear layers, each with 128 units. They compute the mean (μ) and log variance ($log(\sigma^2)$) of a normal distribution. The decoder mirrors the encoder, but replaces downsampling with upsampling. This symmetric enables efficient mesh reconstruction. The network is trained using the Adam optimization algorithm with an initial learning rate of 0.001, which decays linearly with a factor of 0.99, and a batch size of 128 meshes is used during training. To balance reconstruction accuracy and latent space complexity, the KL divergence term is weighted by a factor of 0.005. We minimize the following loss function:

$$\ell_M = MSE(M - \hat{M}) + \beta_M \,\mathrm{KL}\left(f\left(Z_{Mesh} \mid M\right) \| \mathcal{N}(0,1)\right), \tag{A1}$$

where f is the encoder of the mesh β-VAE, β_M is the weight of the KL divergence loss to control the influence of that term on the overall loss. Z_{Mesh} is the sampled latent vector given the input mesh M_i. The mean-squared error (MSE) between the original and reconstructed meshes is calculated, considering all 3D vertices and batch samples. We implement early stopping and restrict training to 300 epochs to mitigate overfitting.

A.2 Image Network

Our image β-VAE model employs a convolutional neural network (CNN) architecture, comprising ResNet blocks, to learn a robust and compact representation of input images. The normalized input images have dimensions of $320 \times 240 \times 3$. The encoder consists of multiple ResNet blocks with varying layer configurations: $[3, 3, 5, 5, 3, 3]$, each followed by downsampling layers with factors of 2, 2, 2, 2, and 5. This hierarchical design allows the model to capture a wide range of features. The decoder mirrors the encoder's architecture, replacing downsampling with upsampling to reconstruct the original image. Throughout the network, we employ the *tanh* activation function. For hyperparameter settings, we annealed β over 50 epochs to a final value of 0.001. The latent dimension Z_{Image} is fixed at 256, allowing for a compact representation of the input images. We initialize the learning rate to 0.001 and apply linear decay with a factor of 0.99, and weight decay with factor 0.00001, which help stabilize training and prevent overfitting. We train our model for up to 300 epochs, implementing early stopping to monitor performance on the validation set and prevent overfitting. This setup enables our image β-VAE model to effectively learn a compact representation of the input images while preserving their essential features. We minimize the following loss function:

$$\ell_I = MSE(I - \hat{I}) + \beta_I \,\mathrm{KL}\left(g\left(Z_{Image} \mid I\right) \| \mathcal{N}(0,1)\right), \tag{A2}$$

where g is the encoder of the image β-VAE and β_I is the weight of the KL divergence loss.

A.3 Projection Network

We train two distinct latent space projection networks: one for projecting meshes' latent space vectors to images' latent space vectors (Mesh-to-Image) and one for the reverse (Image-to-Mesh). Both networks are implemented as Multilayer Perceptrons (MLPs), consisting of 4 layers. The number of neurons in each layer is as follows: [512, 1024, 1024, 256], utilizing the ELU activation function. Dropout regularization is applied after the first two layers, with dropout rates of 0.2 and 0.4, to prevent overfitting. This strategy helps the networks to learn robust representations of the input data. Both networks use a batch size of 512 and minimize the MSE between the predicted and the target values. The architecture of our latent space projection networks is designed to effectively capture the complex relationships between meshes and images, allowing for accurate projections and reconstructions.

A.4 Force Prediction Network

We utilize a unified architecture for all force estimation tasks presented in this paper. The network is implemented as a four-layer MLP with layer sizes of [128, 512, 512, 128]. We employ the *ELU* activation function across all layers. To prevent overfitting and encourage robust representation learning, dropout regularization is applied after the first two layers with a rate of 0.2. The network is trained using a batch size of 512 and a learning rate of 0.001, minimizing the MSE between the predicted and ground-truth force vectors.

Appendix B Analysis of Visual Artifacts in GelSight Reconstruction

While the cross-sensor generalization to GelSight R1.5 demonstrates strong structural alignment, closer inspection of some of the generated samples reveals artifacts resembling surface ripples (see Figure B1, third row). In real-world GelSight imagery, such ripples are typically caused by lateral shear forces deforming the elastomeric coating (Lengiewicz et al. 2020). However, in our generated results, the orientation and magnitude of these ripples do not strictly align with the applied shear forces. We attribute this discrepancy to the distribution of the training data used for fine-tuning. The *ObjectFolder-Real* dataset (Gao et al. 2023) contains recordings rich in shear-induced deformations. Since our fine-tuning process relies on a β-VAE to adapt the decoder to this visual domain, the model learns to reconstruct these textures as intrinsic features of the GelSight style. Consequently, the network hallucinates these ripple patterns to match the statistical distribution of the target domain, rather than explicitly deriving them from the input physics. While this enhances the photorealism of the texture, the ripples should be interpreted as a learned dataset bias rather than a physically accurate simulation of shear dynamics in this specific transfer setting.

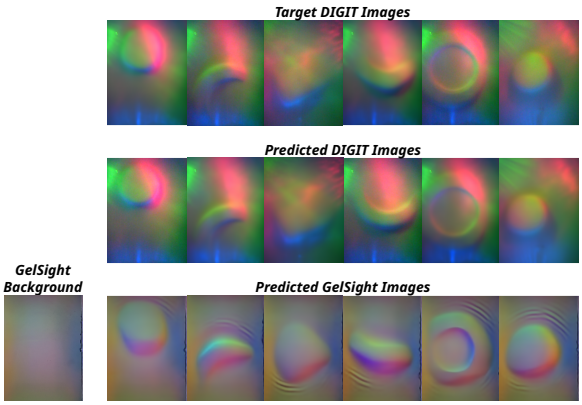

Fig. B1 Visual analysis of domain-specific artifacts. First row: Ground-truth images in the DIGIT domain. Second row: Predicted contact images in the DIGIT domain. Bottom row: Cross-sensor predictions projected into the GelSight R1.5 domain, showing characteristic texture ripple patterns on the surface.

References

Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering. In: International Conference on Neural Information Processing Systems (NIPS), vol. 30th, pp. 3844–3852 (2016)

Gao, R., Dou, Y., Li, H., Agarwal, T., Bohg, J., Li, Y., Fei-Fei, L., Wu, J.: The Objectfolder Benchmark: Multisensory Learning with Neural and Real Objects. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 17276–17286 (2023)

Lengiewicz, J., de Souza, M., Lahmar, M.A., Courbon, C., Dalmas, D., Stupkiewicz, S., Scheibert, J.: Finite Deformations Govern the Anisotropic Shear-Induced Area Reduction of Soft Elastic Contacts. J Mech Phys Solids **143**, 104056 (2020) https://doi.org/10.1016/j.jmps.2020.104056