Haptic Neural Fields: Bringing Tactile Interactions to 3D Rendered Scenes

Supplementary Material

7. Cross-sensor association

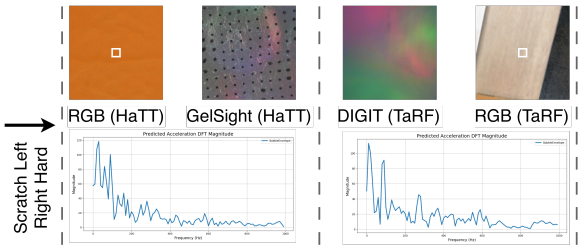

Figure 6. Cross-sensor association. A DIGIT query retrieves a visually similar GelSight map; training HNF on either domain yields comparable vibration spectra for the same action.

A central challenge in our setting is that no single dataset jointly provides (i) GelSight and DIGIT tactile maps, (ii) paired RGB/depth renderings from reconstructed scenes, and (iii) time-domain vibrotactile acceleration signals under diverse actions. As a result, we bridge GelSight and DIGIT through a shared contrastive embedding space trained on *RGB–tactile* pairs from multiple sources (Sec. 5.1).

Shared embedding via RGB anchoring. We train a symmetric contrastive objective on two kinds of pairs: GelSight–RGB (from HaTT/TnG) and DIGIT–RGB (from TaRF). Both sensors are thus aligned *through their visual correspondences* to RGB, encouraging sensor-invariant clustering: tactile observations that correspond to visually similar materials are pulled together, even if their raw tactile appearances differ across sensors (illumination, elastomer, resolution, and field of view).

Retrieval procedure. After training, we embed tactile maps with cosine-normalized features and associate samples across sensors by nearest neighbors in the learned space. Concretely, given a query tactile map (e.g., DIGIT), we compute its embedding and retrieve the closest GelSight embedding among the candidate pool (and vice versa). We use this retrieval to augment HaTT (RGB+GelSight+vibration) with a corresponding DIGIT patch from TaRF, yielding RGB+GelSight+DIGIT triplets used in Sec. 5.2.

Evidence for sensor invariance. Table 2 reports material-classification improvements when the encoders are pretrained on *mixed sensors* and evaluated *per sensor*. This protocol probes invariance (generalization across sensor domains) rather than single-domain feature quality: gains obtained when training with GelSight + DIGIT and testing on either GelSight or DIGIT indicate that the representation is not tied to one sensor format.

Qualitative nearest-neighbor consistency. Figure 6 provides qualitative support for the above: the nearest GelSight map retrieved for a DIGIT query shares consistent texture cues (e.g., repeating micro-structures and scratch patterns). Importantly, HNF models trained on each domain produce very similar spectra for the same action, suggesting that the retrieved association preserves the material-relevant factors needed for downstream signal synthesis. Direct *numerical* retrieval accuracy is difficult to report because TaRF does not release material labels; we therefore complement the cross-sensor probing in Table 2 with the nearest-neighbor visualization in Fig. 6.

8. Geometric sensitivity

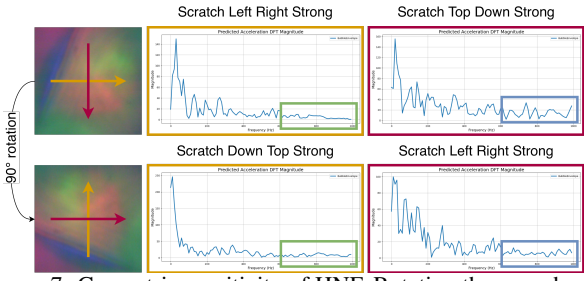

Figure 7. Geometric sensitivity of HNF. Rotating the same haptic map by 90° and swapping to complementary scratch actions (top–down vs. left–right) yields orientation-consistent outputs.

The haptic-map token m is designed to encode both material appearance and local surface geometry: our diffusion model is conditioned on *RGB and depth* to produce a spatially structured haptic map I, which is then encoded into m (Sec. 3.1). This design allows m to capture not only "what material" is present but also geometric cues that affect directional responses (e.g., grooves, brushed patterns, or local relief).

Our main paper already shows location- and direction-dependent outputs (Fig. 4–5), where changing the exploration direction on the same region yields different spectra, consistent with anisotropic tactile responses. Here, Fig. 7 isolates this effect: we rotate the *same* haptic map by 90° and pair it with complementary scratch actions (top–down vs. left–right). HNF produces outputs that remain consistent with the *relative* orientation between action direction and the spatial structure encoded in the map, indicating that

the model leverages geometric/orientation cues rather than relying only on a global material label.

Generalization beyond TaRF scenes. Our pipeline is not intrinsically tied to TaRF scenes. TaRF provides DIGIT patches registered to 3D reconstructions, which is essential for cross-sensor bridging and validation. However, at inference time HNF only requires renderable RGB/depth views (to produce a haptic map and its token) plus an action trajectory/force specification. Therefore, the same pipeline applies to any NeRF/3DGS scene where RGB and depth can be rendered, and where the corresponding material class (or a close proxy) is supported by the trained HNF library.

9. Mathematical clarification of HNF

For clarity, we provide below a more detailed analysis of the mathematical formulation behind the HNF.

For a contact segment S_c, we assume the haptic map I to be locally constant and define the conditioning token

$$\mathbf{m} = E(I).$$

Let $\mathbf{u}(t) = [\mathbf{d}(t),\, v(t),\, F_z(t)]$ be the action state introduced in Sec. 3. For each output instant t_i, we define a causal temporal ray over a short history window:

$$\mathbf{r}_i(\tau) = \big(\mathbf{m},\, \mathbf{u}(t_i - \tau)\big), \qquad \tau \in [0, H],$$

where H is the temporal horizon. The HNF of Eq. 1 is then evaluated along this ray as

$$F_\Theta(\mathbf{r}_i(\tau)) = \big(\alpha_i(\tau),\, \sigma_i(\tau)\big),$$

where $\alpha_i(\tau) \in \mathbb{R}$ is the local emitted acceleration contribution and $\sigma_i(\tau) \in \mathbb{R}_{\geq 0}$ is the temporal density.

Similarly to NeRF, we define the transmittance along the temporal ray as

$$T_i(\tau) = \exp\left(-\int_0^\tau \sigma_i(s)\, ds\right),$$

and synthesize the acceleration at time t_i by compositing the local contributions:

$$\hat{a}(t_i) = \int_0^H T_i(\tau)\, \sigma_i(\tau)\, \alpha_i(\tau)\, d\tau.$$

Using the discrete temporal neighborhood $\mathcal{N}_i = \{t_{i,n}\}_{n=1}^N$ from Sec. 3.2.2, with $t_{i,n} = t_i - (n-1)\Delta t$, this becomes

$$(\alpha_{i,n}, \sigma_{i,n}) = F_\Theta\big(\mathbf{m}, \mathbf{u}(t_{i,n})\big),$$

$$T_{i,n} = \exp\left(-\sum_{j<n} \sigma_{i,j} \Delta t\right),$$

$$w_{i,n} = T_{i,n}\big(1 - \exp(-\sigma_{i,n}\Delta t)\big),$$

which recovers Eq. 2, and yields

$$\hat{a}(t_i) = \sum_{n=1}^{N} w_{i,n}\alpha_{i,n},$$

as in Eq. 3. Hence, $\hat{a}(t_i)$ depends on the recent history of the action through the causal window $\mathcal{N}_i$, while $\mathbf{m}$ provides the local surface context held fixed over the segment.

10. Further discussions

Dataset augmentation. Our augmentation strategy aimed to generate pseudo-labeled samples to expand the HNF training set. First, we defined novel exploratory actions, namely new combinations of force, speed, and direction. To this end, we provided ChatGPT-5 with example trajectories and asked it to generate the eight motion patterns described in Sec. 5.3, following the same structure.

Specifically, we used the files `Position_Carpet 2.xml` and `Force_Carpet 2.xml` from the HaTT dataset, with the following query:

```
Given these two files, can you create
different actions which consist of 1000
values for each property? Generate the
following 8 human-like actions:
```

- `scratching left to right pressing strong`
- `scratching left to right pressing weak`
- `scratching top to bottom pressing strong`
- `scratching top to bottom pressing weak`
- `scratching along one diagonal pressing strong`
- `scratching along another diagonal pressing weak`
- `rubbing randomly moving fast`
- `rubbing randomly moving slow`

Second, we used the original HNF models, trained on ground-truth signals, to infer synthetic acceleration spectra for these new actions while preserving the material-specific DIGIT observations as visual conditioning. For each material, the model combined the corresponding DIGIT images with the force, speed, and direction signals of the new actions to synthesize novel acceleration samples in the same DFT domain used during training. This process increased the diversity of action-conditioned training samples.

From semantic actions to closed-loop interaction. In the paper we report results using canonical action primitives

for controlled evaluation and reproducibility. The formulation itself is compatible with continuous, real-time sensorimotor inputs: a tracked tool/finger provides the contact trajectory $p(t)$ and normal force $F_z(t)$, from which we compute $u(t) = [d(t), v(t), F_z(t)]$ (Sec. 3). In a deployed system, these quantities can be streamed to HNF to synthesize vibration samples online.

End-to-end latency considerations. End-to-end latency depends on three components: (i) tracking (pose/force sensing), (ii) scene querying (rendering RGB/depth and producing a haptic map/token), and (iii) haptic synthesis + actuation. Step (ii) can be amortized by caching predicted haptic maps/tokens per surface region or by precomputing them for static scenes. Step (iii) is lightweight (MLP inference with short-window temporal compositing) and naturally supports streaming. Practical real-time performance will therefore be primarily driven by the sensing and actuation stack, and by how aggressively the scene-side computations are cached.

Perceptual validation. We evaluate waveform fidelity with objective signal metrics (ST-SIM, LSD, MSE) and show qualitative direction-dependent behavior. A natural next step is a controlled user study on a vibrotactile device (e.g., stylus-based or wearable), comparing (a) recorded vs. synthesized signals and (b) signals generated for different materials/actions. Suitable protocols include ABX discrimination, material identification, and preference/realism ratings under matched action conditions.

Limitations: data/hardware vs. model. Data limitations include the lack of fully aligned multi-sensor, multi-action datasets under a single calibration, motivating our contrastive bridging and per-material training. Hardware limitations include actuator bandwidth, amplitude limits, and the challenge of reproducing multi-axis stimuli on commodity devices. Model-side limitations include sensitivity to association errors and reliance on the set of materials seen during training; both can be mitigated by richer datasets, improved cross-sensor alignment, and scaling the material library.