GROUNDING VISION, TOUCH, AND SOUND FOR CONTACT-RICH DEXTEROUS MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision, touch, and sound provide complementary cues for contact-rich dexterous manipulation, yet their distinct spatial and temporal structures make effective fusion challenging. Moreover, requiring every demonstration to contain all modalities limits the amount of usable training data. We introduce **DexMuse**, a plug-in multisensory fusion module that anchors touch, sound, and local vision at the fingertips of a dexterous hand, where contact most often occurs. Calibrated fingertip projections associate local visual features with tactile and proprioceptive context, while the resulting region tokens attend to acoustic history and complement global visual observations. We further propose ***availability-aware mixed training***, which lets demonstrations missing touch or sound supervise the same policy without placeholder tokens or synthesized signals. We integrate DexMuse into both ACT and the pretrained vision-language-action model $\pi_{0.5}$ through backbone-specific interfaces. Across seven challenging real-world single-arm and bimanual tasks, DexMuse reaches average success rates of 0.70 with ACT and 0.80 with $\pi_{0.5}$, compared with 0.35 and 0.52 for concatenation-based fusion, and outperforms all baselines on every task. At a fixed budget of complete demonstrations, adding incomplete demonstrations improves DexMuse by 0.19 on average, more than twice the gain of concatenation-based fusion. Real-world videos are available at dexmuse-submission.github.io.

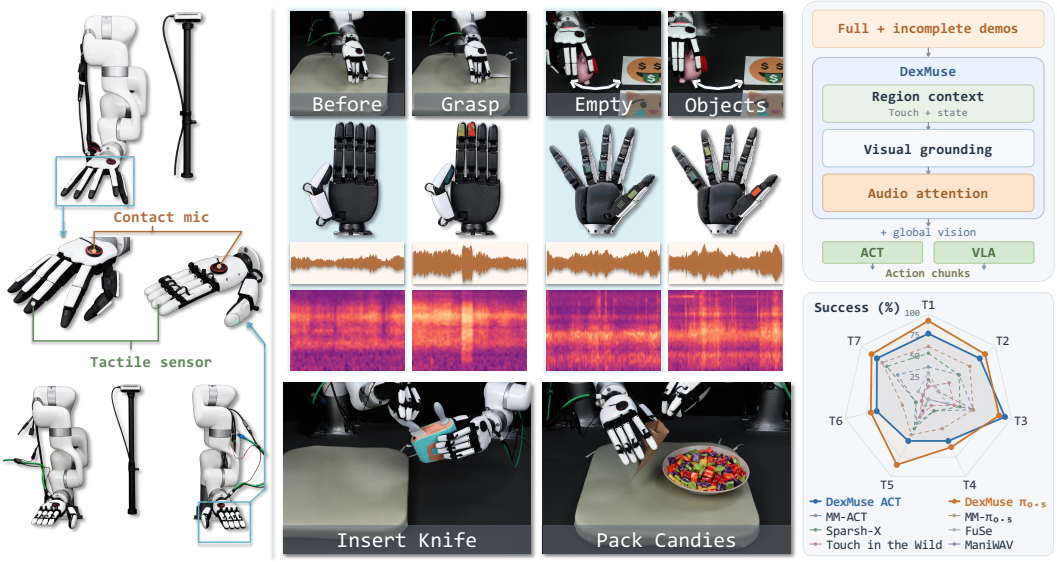


Figure 1: **DexMuse overview.** Left: single-arm and bimanual platforms with contact microphones and tactile sensors. Middle: audio and touch reveal interaction cues when visual changes are subtle or occluded. Top right: DexMuse fuses vision, touch, and audio and learns from complete and incomplete demonstrations. Bottom right: success rates (%) on all tasks.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Dexterous manipulation requires inferring interaction states that images alone may not reveal, such as contact, sensed through touch, and hidden events, heard through sound (Jing et al., 2026). Inserting a stopper involves both alignment and contact, while shaking an opaque container can provide acoustic cues about its hidden contents. Vision, touch, and sound thus provide complementary cues about object configuration, local contact, and hidden events (Liu et al., 2024; Zhu et al., 2025; Jiang et al., 2026b) (Figure 2). For a dexterous hand, these cues meet at the fingertips: tactile sensors are typically concentrated there, fingertips are where contact most often occurs, and contact in turn produces both tactile responses and interaction sounds. The fingertips thus offer a natural place to bring the three modalities together.

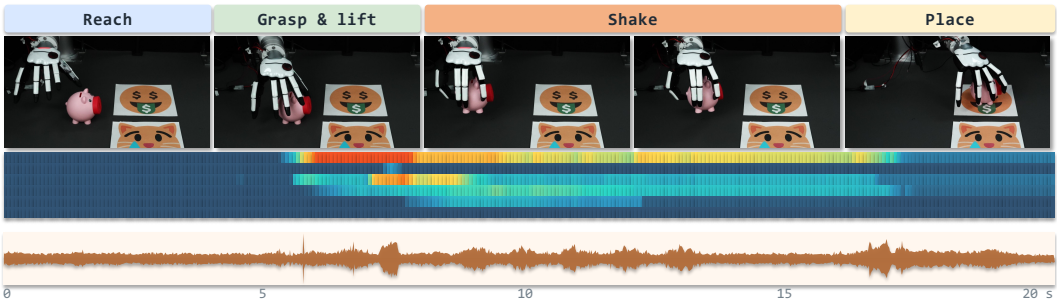


Figure 2: **Complementary sensory observations during piggy-bank manipulation.** RGB frames show the object, hand configuration, and target locations. Tactile heatmap rows, ordered as thumb, index, middle, ring, little finger, and palm, show changes in regional responses relative to the initial baseline. Acoustic peaks accompany five shakes beginning around 8 s, providing evidence of motion inside the opaque container. RGB frames illustrate phases from a separate demonstration and are not synchronized with the sensor traces.

Yet learning dexterous policies from all three modalities remains largely unexplored (Table 1). Policies that combine vision, touch, and sound have mostly used parallel grippers (Li et al., 2022; Feng et al., 2024; Zhao et al., 2025; Jiang et al., 2026a), while dexterous-hand policies typically pair vision with touch (Guzey et al., 2023; Lin et al., 2025; Luo et al., 2026; Ye et al., 2025) or fuse touch and audio into fingertip representations whose backbone excludes scene vision (Higuera et al., 2025). These policies process each modality as a separate stream, combined by attention, concatenation, or stage-dependent weighting (Li et al., 2022; Zhao et al., 2025; Feng et al., 2024), and leave the policy to discover where touch and sound originate. Luo et al. (2026) comes closest by aligning tactile measurements with the image, but it includes neither sound nor bimanual control. None of these approaches anchors touch, sound, and local vision at a common location on the hand.

Collecting complete multisensory demonstrations is also expensive, while demonstrations that miss touch or sound still carry useful action supervision. Existing approaches either synthesize the missing signals (Wang et al., 2025; Li et al., 2026c) or handle an absent stream as a whole (Ma et al., 2022), replacing it with a learned token (Clark et al., 2026) or masking it (Octo Model Team et al., 2024). Synthesis introduces a generative model whose errors enter training, and stream-level handling discards the structure that the missing modality shares with the rest of the observation. Anchoring touch and sound at the fingertips avoids both: kinematics and calibrated vision locate every fingertip in every demonstration, so each region always forms a token that touch and sound only augment. A missing modality removes only that evidence, with nothing generated or discarded.

We introduce **DexMuse**, a plug-in module built on this fingertip anchor (Figure 1). Each finger, and the palm when instrumented, forms a hand region anchored at its fingertip. Calibrated projection locates the fingertip in the image to retrieve local visual features, the region's tactile sensors add contact evidence, and the resulting region token queries the hand's recent sound, while global image tokens retain scene context. When touch or sound is not recorded, availability-aware mixed training masks only that evidence, so incomplete demonstrations still train the same region tokens and action predictor. We integrate DexMuse into ACT (Zhao et al., 2023), a policy learned from task demonstrations, and into the pretrained vision-language-action model $\pi_{0.5}$ (Physical Intelligence et al., 2025), testing the module with both a task-specific policy and a pretrained generalist. Across seven real-world single-arm and bimanual tasks, DexMuse reaches average success rates of 0.70

Method	Arms	Vision	Active audio	Tactile	Dexterous hand	Multi-EEF validation	Incomplete demonstrations
MULSA (Li et al., 2022)	Single	✓	✗	✓	✗	✗	✗
ManiWAV (Liu et al., 2024)	Single	✓	✓	✗	✗	✗	✗
MS-Bot (Feng et al., 2024)	Single	✓	✗	✓	✗	✗	✗
PolyTouch (Zhao et al., 2025)	Bimanual	✓	✓	✓	✗	✗	✗
Sparsh-X[1] (Higuera et al., 2025)	Single	✗	✓	✓	✓	✗	✗
RGB-S (Luo et al., 2026)	Single	✓	✗	✓	✓	✗	✗
FuSe (Jones et al., 2025)	Single	✓	✓	✓	✗	✗	✗
MultiGen (Wang et al., 2025)	Single	✓	✓	✗	✗	✗	✓
MuSe (Clark et al., 2026)	Single	✓	✗	✓	✗	✗	✓
DexMuse (Ours)	**Both**	✓	✓	✓	✓	✓	✓

Table 1: **Comparison of selected methods.** Arms reports the evaluated configurations; Both denotes single-arm and bimanual setups. Multi-EEF validation requires real-world evaluations of the multimodal manipulation method on at least two distinct end-effector designs, excluding data collection and perception-only tests. Active audio uses robot- or hand-mounted microphones.

with ACT and 0.80 with $\pi_{0.5}$, compared with 0.35 and 0.52 for concatenation-based fusion on the same backbones, and outperforms every baseline on every task. Adding incomplete demonstrations at a fixed complete-demonstration budget improves DexMuse by 0.19 on average, more than twice the gain of concatenation-based fusion (0.08).

Our contributions are:

- A plug-in module that anchors local vision, tactile feedback, and acoustic history at the fingertips of a dexterous hand, complementing global scene observations.
- Availability-aware mixed training, which masks missing touch or sound within each hand region while its fingertip anchor of kinematics and calibrated vision remains intact.
- Real-world experiments on seven single-arm and bimanual tasks with two dexterous hands, in which DexMuse outperforms all baselines on every task (0.70 vs. 0.35 with ACT and 0.80 vs. 0.52 with $\pi_{0.5}$ over concatenation) and gains more from incomplete demonstrations (0.19 vs. 0.08).

2 Related Work

2.1 Multisensory manipulation policies

Audio and touch expose interaction states that images may miss. Du et al. (2022) and Liu et al. (2024) condition imitation policies on images and sound, Mejia et al. (2024) transfers pretrained audio-visual representations to contact-microphone observations, and Nie et al. (2026) retains transient acoustic events through streaming causal audio memory. Touch in the Wild learns fine-grained manipulation with a portable visuotactile gripper (Zhu et al., 2025). Multifingered visuotactile policies build on tactile pretraining in Guzey et al. (2023), bimanual imitation in Lin et al. (2025), and image-aligned tactile saliency in Luo et al. (2026). Sound also aids dexterity by supervising grasp gentleness (Nakahara & Calandra, 2025) or estimating contact and slip from vibro-acoustic signals (Mao et al., 2026).

Policies combining all three modalities have mostly used parallel grippers: Li et al. (2022) applies self-attention over vision, touch, and sound, Feng et al. (2024) adjusts their fusion by task stage, Zhao et al. (2025) conditions diffusion policies on them in bimanual tasks, and Jones et al. (2025) grounds heterogeneous sensors in language to fine-tune generalist policies. Yu et al. (2024) learns from human tactile, acoustic, and motion demonstrations. For multifingered hands, Higuera et al. (2025) fuses tactile images, audio, motion, and pressure into fingertip representations for Allegro-hand control, with scene vision supplied separately. We instead study coordinated arm–finger control in single-arm and bimanual settings, where scene vision, distributed touch, and interaction sounds condition one policy (Table 1).

[1]We compare fusion-module inputs; Sparsh-X's downstream wrist-camera input is excluded.

2.2 Cross-modal correspondence

Vision–touch methods associate appearance with local contact through attention (Chen et al., 2022; Heng et al., 2026), patch alignment (Li et al., 2026a), shared 3D spaces (Huang et al., 2024), or geometric projection (Luo et al., 2026; Van der Merwe et al., 2026; Liu et al., 2026b), and hand-aware representations preserve canonical regions or hand-pose context (Zhang et al., 2025; 2026b; Yuan et al., 2026). Audio-visual policies use sound without an explicit link to distributed touch (Liu et al., 2024; Nie et al., 2026); these pairwise links do not specify how local contact should retrieve acoustic evidence. We use hand regions as a shared physical reference: fingertip projections link local vision to tactile and proprioceptive context, and region tokens query acoustic history.

2.3 Policy learning from heterogeneous data

Octo masks unavailable observation tokens across datasets (Octo Model Team et al., 2024), and missing-modality learning in vision and language handles absent streams via modality dropout (Neverova et al., 2016), reconstruction (Ma et al., 2021), robust fusion (Ma et al., 2022), or prompts (Lee et al., 2023). Other approaches synthesize missing audio (Wang et al., 2025), tactile signals (Li et al., 2026c; Zhang et al., 2026b), or grounded sensor images (Liu et al., 2026a). Closest to ours, MuSe jointly fine-tunes on new and original demonstrations with learned tokens for missing force–torque inputs (Clark et al., 2026). These methods handle a missing stream as a whole; availability-aware mixed training instead masks missing touch or sound within each hand region, whose fingertip anchor remains observed. Appendix F extends this discussion.

3 Method

DexMuse is a plug-in module that converts vision, touch, and acoustic history into hand-region interaction tokens (Figure 3). Backbone-specific interfaces use these tokens to condition actions. The hand supplies a common physical reference for fusion and a consistent region layout for learning from demonstrations with different sensory coverage.

Problem setting. At time t, a policy receives images I_t, robot state q_t, and available touch T_t and acoustic history S_t, and predicts an action chunk $\mathbf{a}_t = (a_t, \dots, a_{t+H-1})$ for arm motion and hand joints. The VLA also receives a task instruction c. Training combines complete demonstrations $\mathcal{D}_{\text{full}}$ and incomplete demonstrations $\mathcal{D}_{\text{inc}}$, all retaining images, states, and actions with compatible geometry, calibration, and action conventions within a policy instance. An episode-level mask $m = (m_T, m_S) \in \{0,1\}^2$ specifies sensory availability: $m = (1,1)$ for complete records and $m \neq (1,1)$ otherwise. Deployment uses all three modalities.

3.1 Hand-region multisensory fusion

Each active hand $h \in \mathcal{H}$ has regions $\mathcal{R}_h$ for its fingers and, when instrumented, palm, with associated tactile measurements, audio, and a calibrated view. The module produces $N = \sum_{h\in\mathcal{H}} |\mathcal{R}_h|$ tokens. The visual backbone provides spatial features F_t and global tokens V_t. Touch and sound enter each region as evidence added to an anchor built from proprioception and calibrated vision; we first describe complete inputs, and Section 3.3 uses this structure for incomplete demonstrations.

Hand configuration and touch establish regional context. Calibrated projection then supplies the fingertip-to-image correspondence, and the enriched regions query acoustic history. Geometry provides spatial association; attention learns how the current interaction context uses sound.

Region context. We suppress time and hand indices below. For region r, E_P combines its associated arm and hand state q_r with a learned region identity, while E_T aggregates its tactile map or grouped force measurements. Encoders are shared across hands of the same type within a policy instance (Appendix A.1). Their d-dimensional features define

$$b_r = \mathrm{LN}(\tau_r + \rho_r), \qquad \tau_r = E_T(T_r), \quad \rho_r = E_P(q_r, r). \tag{1}$$

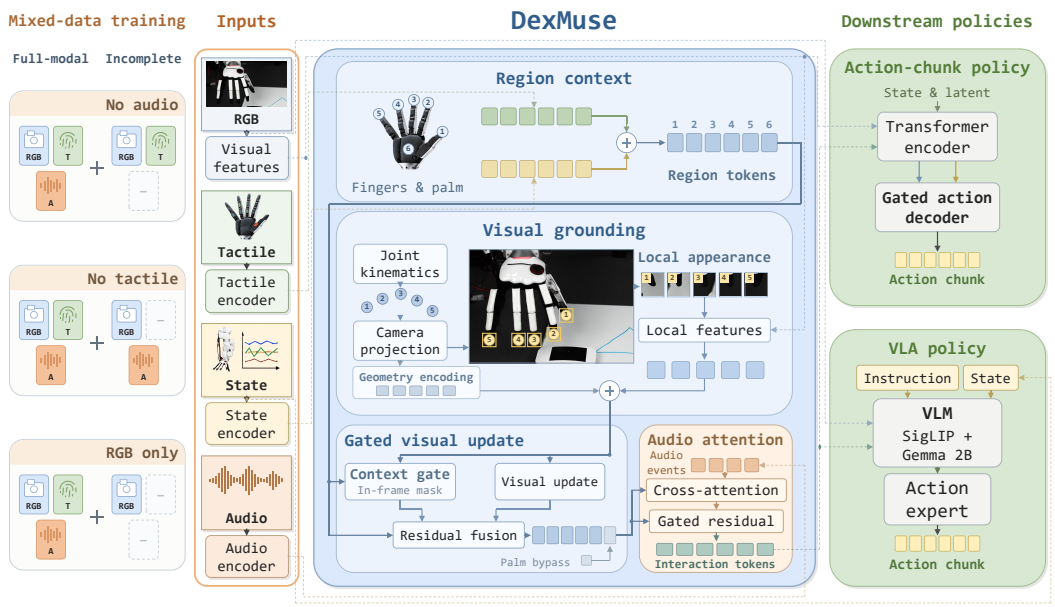


Figure 3: **Hand-region fusion and availability-aware training in DexMuse.** Complete and incomplete demonstrations share an observation interface and action supervision. Touch and state initialize hand-region tokens; calibrated local vision enriches these tokens before they query audio history. The resulting interaction tokens and global vision condition ACT or $\pi_{0.5}$.

Geometric visual association. Forward kinematics and camera calibration map each fingertip to a camera-frame point x_r and image location p_r. Physical joint angles are aligned to the image timestamp, and image transformations are applied to the sampling coordinates. We retrieve a local visual feature and encode its geometric context:

$$\begin{aligned} \bar{v}_r &= \mathrm{LocalPool}(F, p_r), \\ v_r &= W_V \bar{v}_r + E_G([\tilde{p}_r; x_r; \nu_r]). \end{aligned} \tag{2}$$

Here LocalPool combines valid bilinear samples from a 3×3 neighborhood with fixed Gaussian weights. The geometry encoder E_G uses normalized image coordinates $\tilde{p}_r$, camera coordinates x_r, and a flag ν_r indicating positive depth and an in-image projection. A learned gate controls the visual residual:

$$z_r = \mathrm{LN}\big(b_r + \nu_r g_r^V U_V(v_r)\big), \qquad g_r^V = \sigma(w_V^\top [b_r; v_r] + \beta_V). \tag{3}$$

The projection U_V is gated using regional context and local appearance. Fingertip neighborhoods provide visual context for each finger's aggregated tactile state. Palm regions bypass this step, retaining $z_r = b_r$, while global tokens V_t retain wider object and target context. Sampling and initialization details appear in Appendix A.2.

Region-conditioned acoustic integration. Each hand's audio input covers a six-second history ending at the stream-corrected observation timestamp. A temporal encoder produces $L = 4$ ordered interval summaries, $A = E_A(S) \in \mathbb{R}^{L \times d}$, with learned temporal-position embeddings. The current region context queries this history through multi-head cross-attention:

$$c_r = \mathrm{MHA}(z_r, A, A), \qquad h_r = \mathrm{LN}\big(z_r + g_r^A c_r\big), \tag{4}$$

where $g_r^A = \sigma(w_A^\top [z_r; c_r] + \beta_A)$ modulates the acoustic update. Regional queries can weight the same fixed acoustic history differently. Tokens from all active hands retain region and hand identities and are processed jointly by the policy. Audio and identity encodings are detailed in Appendix A.1.

3.2 Integration with Policy Backbones

DexMuse outputs region tokens alongside global image tokens and connects to a backbone through a light interface that keeps the backbone's action objective (Appendix A.3). For ACT (Zhao et al.,

2023), a frozen DINOv2 encoder (Oquab et al., 2024) supplies the spatial features for fingertip sampling and the global image tokens; region tokens, robot state, and the ACT latent form an interaction memory, global image tokens form a scene memory, and each decoder layer attends to the interaction memory before a gated cross-attention update from the scene memory. Training keeps ACT's ℓ_1 reconstruction and KL objectives. For $\pi_{0.5}$ (Physical Intelligence et al., 2025), the native SigLIP encoder supplies the spatial features, and region tokens are projected to the language-model width and inserted between the image and language tokens of the observation prefix; pretrained and fusion parameters are fine-tuned jointly with the original flow-matching objective and action sampler.

3.3 Availability-aware mixed training

Each region token consists of a fingertip anchor that every demonstration records, namely proprioception, region identity, and calibrated local vision, plus touch and sound evidence attached to it. When an episode lacks touch or sound, we mask only that evidence. With the episode mask $m = (m_T, m_S)$, Eqs. 1 and 4 become

$$b_r = \mathrm{LN}(\rho_r + m_T\,\tau_r)\,, \qquad h_r = \mathrm{LN}\big(z_r + m_S\,g_r^A\,c_r\big)\,, \tag{5}$$

while Eq. 3 is unchanged. Complete and incomplete records share one token layout; a zero-contact reading or a quiet recording is still encoded, whereas a missing stream adds nothing. Because masked outputs receive no gradient, an incomplete record trains the anchor encoder, the fingertip visual association, acoustic attention when sound is recorded, and the action predictor. We train one policy on $\mathcal{D}_{\text{full}} \cup \mathcal{D}_{\text{inc}}$ with the unchanged backbone objective, drop each hand's visual and acoustic residuals with probability 0.1 during training, fix $\mathcal{D}_{\text{full}}$, and deploy with $m = (1, 1)$.

4 Experiments

We evaluate DexMuse on real robots to answer three questions: whether fusion around the hand improves multisensory dexterous manipulation over multimodal and published baselines, whether vision, touch, and sound provide complementary evidence and how hand-region fusion changes the learned representation, and whether additional incomplete demonstrations improve a policy at a fixed complete-demonstration budget, with larger gains for hand-region fusion than for concatenation. Sections 4.2–4.4 address these questions in turn.

4.1 Hardware platforms and tasks

All experiments run on UFACTORY xArm7 arms, which are 7-DoF manipulators, in the single-arm and bimanual configurations shown in Figure 4. Each arm carries a dexterous hand: either the 12-DoF XHand, whose tactile sensing is organized into five finger regions, or the XHand Pro, whose 18 tactile sensors cover the five fingers and the palm; Appendix B describes both platforms. A contact microphone on the back of each hand records interaction sounds, and an external RGB camera calibrated to the robot base provides the view used for fingertip projection.

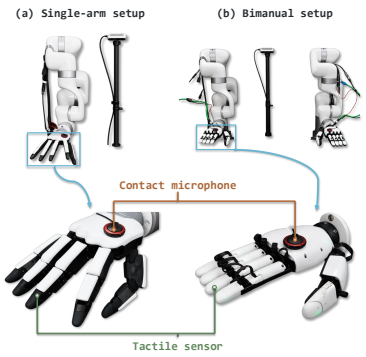


Figure 4: **Hardware setups.**

Figure 5 shows the seven tasks: three single-arm tasks that involve tool use and content-dependent sorting, and four bimanual tasks that involve insertion, packing, and pouring. Each panel shows representative frames and the task objective. Policies are trained from teleoperated demonstrations; Appendices C and D describe data collection and training.

4.2 Multisensory dexterous manipulation

Table 2 compares DexMuse with two groups of baselines. Within each policy family, the multimodal baselines MM-ACT and MM-$\pi_{0.5}$, where MM stands for multimodal, receive the same vision, touch, sound, and robot-state inputs and use the same image encoder as DexMuse, but concatenate per-modality tokens instead of fusing them around the hand. The published methods provide reference

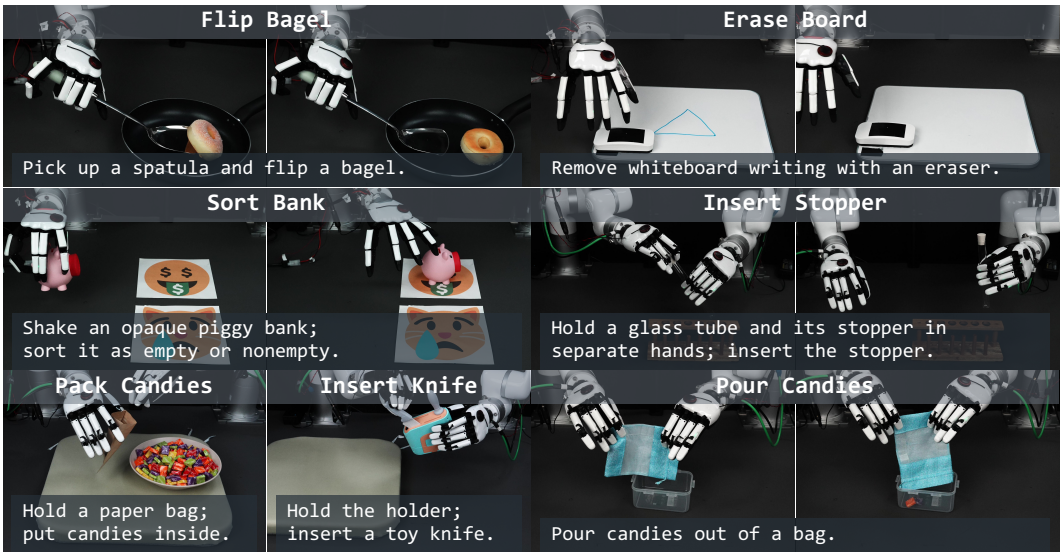


Figure 5: **Seven real-world manipulation tasks.** Single-arm tasks and bimanual tasks, each shown with representative frames and its objective.

designs: ManiWAV for audio-visual policies, Touch in the Wild for visuotactile policies, and Sparsh-X and FuSe for multisensory representations. Appendix E gives the implementation details of all baselines.

DexMuse achieves the highest success rate on every task with both backbones. It raises the average success rate from 0.35 to 0.70 with ACT and from 0.52 to 0.80 with $\pi_{0.5}$. Among the published methods, Sparsh-X performs best at 0.38, while the audio-visual ManiWAV and the visuotactile Touch in the Wild both reach 0.18. The margins are large on bimanual insertion: no baseline exceeds 0.38 on Insert Stopper or 0.31 on Insert Knife, whereas DexMuse reaches 0.54 and 0.62 with ACT and 0.62 and 0.69 with $\pi_{0.5}$. Sort Bank separates the methods most clearly. Placing the piggy bank correctly requires inferring its hidden contents by shaking it, and ManiWAV, MM-ACT, and MM-$\pi_{0.5}$ each succeed in 0.54 of trials, close to chance for the two-way placement decision, whereas DexMuse reaches 0.92 with ACT and 0.85 with $\pi_{0.5}$. The module thus improves both a task-specific policy and a pretrained generalist.

Method	Flip Bagel	Erase Board	Sort Bank	Insert Stopper	Pack Candies	Insert Knife	Pour Candies	Avg.
ManiWAV (Liu et al., 2024)	0.23	0.15	0.54	0.00	0.15	0.08	0.08	0.18
Touch in the Wild (Zhu et al., 2025)	0.15	0.31	0.31	0.08	0.23	0.08	0.08	0.18
MM-ACT (Zhao et al., 2023)	0.38	0.46	0.54	0.15	0.31	0.15	0.46	0.35
MM-$\pi_{0.5}$ (Physical Intelligence et al., 2025)	0.62	0.62	0.54	0.38	0.46	0.31	0.69	0.52
Sparsh-X (Higuera et al., 2025)	0.54	0.46	0.38	0.15	0.38	0.15	0.62	0.38
FuSe (Jones et al., 2025)	0.31	0.15	0.23	0.00	0.08	0.00	0.15	0.13
DexMuse (ACT, Ours)	0.77	0.77	**0.92**	0.54	0.54	0.62	0.77	0.70
DexMuse ($\pi_{0.5}$, Ours)	**0.92**	**0.85**	0.85	**0.62**	**0.85**	**0.69**	**0.85**	**0.80**

Table 2: **Main real-world evaluation.** Success rates over 13 trials per method and task (Figure 5); Avg. is the mean across all seven tasks. MM denotes multimodal adaptations of the corresponding backbone. Bold and underline mark the best and second-best result in each column.

4.3 Complementary sensing and hand-region representations

Table 3 ablates the sensory inputs of DexMuse with ACT, using each input condition for both training and evaluation. Vision alone reaches an average of 0.26, adding touch or sound raises it to 0.45 and 0.47, and combining both reaches 0.70. The two modalities help different tasks. Touch raises Erase

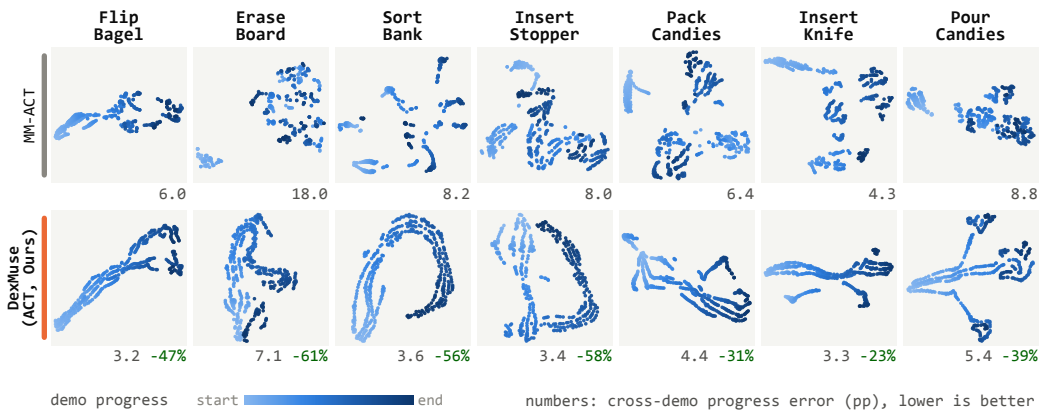


Figure 6: **ACT features at the policy-transformer input.** t-SNE of demonstration frames for MM-ACT and DexMuse (ACT), colored by demonstration progress and rotated so that progress runs left to right. Numbers give the cross-demonstration progress error in percentage points (lower is more consistent) and, for DexMuse, its relative decrease.

Board from 0.31 to 0.46, where maintaining contact with the board matters, and Pour Candies from 0.38 to 0.69, more than sound does on either task. Sound helps most on Sort Bank, where shaking reveals the hidden contents, raising success from 0.38 to 0.77; it also raises Flip Bagel from 0.23 to 0.62 and Insert Knife from 0.08 to 0.31, more than touch does. Combining both exceeds the better single modality on every task, by 0.31 on Erase Board and Insert Knife and by 0.08 on Pack Candies and Pour Candies, so touch and sound contribute complementary rather than redundant evidence.

Input Modalities	Flip Bagel	Erase Board	Sort Bank	Insert Stopper	Pack Candies	Insert Knife	Pour Candies	Avg.
RGB	0.23	0.31	0.38	0.15	0.31	0.08	0.38	0.26
RGB + Tactile	0.46	<u>0.46</u>	0.54	<u>0.31</u>	<u>0.46</u>	0.23	<u>0.69</u>	0.45
RGB + Audio	<u>0.62</u>	0.31	<u>0.77</u>	<u>0.31</u>	<u>0.46</u>	<u>0.31</u>	0.54	<u>0.47</u>
RGB + Tactile + Audio	**0.77**	**0.77**	**0.92**	**0.54**	**0.54**	**0.62**	**0.77**	**0.70**

Table 3: **Modality ablations of DexMuse with ACT.** Each condition specifies the sensory inputs used for both training and evaluation; robot state is retained throughout. Task entries report success rates over 13 trials per task and input condition. Avg. is the unweighted mean across all seven tasks. Bold and underlined entries mark the best and second-best results in each column.

Figure 6 compares MM-ACT and DexMuse with ACT at the input of the policy transformer. Each frame is described by its mean global-visual token together with its mean auxiliary token for concatenation or its mean hand-region token for DexMuse. We embed the frames of four demonstrations per task, or five for Pack Candies, with t-SNE and color them by their progress within the demonstration. With concatenation, the embeddings break into disconnected clusters; with DexMuse, frames from different demonstrations form parallel bands that follow the task from start to end. We quantify this with a cross-demonstration progress error: for every frame, we retrieve its ten nearest frames from the other demonstrations in the original feature space and measure the difference in normalized progress. Across the seven tasks, DexMuse lowers this error by 23% on Insert Knife up to 61% on Erase Board. Appendix A.4 provides the feature-analysis configurations.

Figure 7 shows how the action queries of the two ACT policies attend to the image tokens they receive at the input of the policy transformer, the representation analyzed in Figure 6. We average the cross-attention of the first decoder layer over action queries and heads; as in the original ACT implementation, the action head reads only this layer, so later decoder layers do not affect the predicted actions. On the frames shown, the attention of DexMuse concentrates on the hand that holds the spatula in Flip Bagel and the piggy bank in Sort Bank, reaching 8.8 and 6.3 times the uniform level, whereas the attention of MM-ACT peaks at 1.4 times the uniform level, mostly on the table. Over

100 sampled frames from each of these demonstrations, the median peak ranges from 3.2 to 10.0 times the uniform level for DexMuse across the seven tasks and from 1.4 to 1.7 times for MM-ACT. Figure 8 in Appendix A.2 visualizes the calibrated fingertip projections, the sampled neighborhoods, and the per-finger touch on an Insert Stopper demonstration.

4.4 Learning from additional incomplete demonstrations

We split each task's training data into two disjoint halves of nearly equal size. One half remains complete in every condition, and the other half is added either fully observed or with touch, sound, or both withheld, as listed in Table 4. All policies are evaluated in 13 real-world trials per task with all modalities available, and both fusion designs are trained on identical episodes and availability masks, with the same missing-modality handling and modality dropout, so that they differ only in fusion (Appendix E).

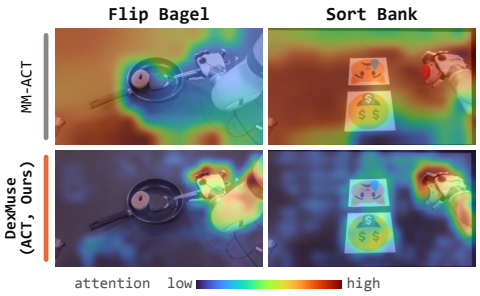


Figure 7: **Attention of the ACT action queries over the input image tokens** in the first decoder layer, the only one read by the action head, on the strongest-touch frame of a demonstration, normalized per map; peaks reach 1.4 times the uniform level for MM-ACT and 6.3–8.8 times for DexMuse.

Adding incomplete demonstrations helps both designs: every mixed condition matches or exceeds the complete-half baseline on every task. The gains are larger for DexMuse. Starting from the complete half, demonstrations without touch raise the average success of DexMuse from 0.25 to 0.48, demonstrations without sound to 0.44, and RGB-only demonstrations to 0.42, whereas MM-ACT rises from 0.19 to 0.29, 0.29, and 0.23. Pooled over the three conditions, DexMuse gains 0.19 and MM-ACT gains 0.08, consistent with incomplete records continuing to train the fingertip visual association and region layout under the anchor-preserving masking of Section 3.3. All mixed conditions remain below the fully observed references of 0.70 for DexMuse and 0.35 for MM-ACT, so the withheld modalities still carry information that incomplete records cannot supply.

Method	Training Data	Flip Bagel	Erase Board	Sort Bank	Insert Stopper	Pack Candies	Insert Knife	Pour Candies	Avg.
MM-ACT	50% Full-modal	0.23	0.23	0.38	0.08	0.15	0.08	0.15	0.19
	50% Full-modal + 50% w/o Tactile	0.31	0.38	0.46	**0.15**	0.23	**0.15**	0.31	0.29
	50% Full-modal + 50% w/o Audio	**0.38**	0.23	0.38	**0.15**	**0.31**	**0.15**	0.38	0.29
	50% Full-modal + 50% RGB-only	0.31	0.31	0.38	**0.15**	0.15	0.08	0.23	0.23
	100% Full-modal	**0.38**	**0.46**	**0.54**	**0.15**	**0.31**	**0.15**	**0.46**	**0.35**
DexMuse	50% Full-modal	0.23	0.23	0.46	0.23	0.15	0.15	0.31	0.25
	50% Full-modal + 50% w/o Tactile	0.38	0.31	0.69	0.46	0.38	0.54	0.62	0.48
	50% Full-modal + 50% w/o Audio	0.38	0.54	0.54	0.38	0.31	0.23	0.69	0.44
	50% Full-modal + 50% RGB-only	0.54	0.23	0.54	0.31	0.46	0.38	0.46	0.42
	100% Full-modal	**0.77**	**0.77**	**0.92**	**0.54**	**0.54**	**0.62**	**0.77**	**0.70**

Table 4: **Using incomplete demonstrations with ACT policies.** MM-ACT is the multimodal ACT baseline of Table 2. Success rates over 13 trials per condition; within each method, bold and underline mark the best and second-best result in each column.

5 Conclusion

We presented DexMuse, which anchors local vision, touch, and acoustic history at the fingertips of a dexterous hand and trains one policy from demonstrations with complete and incomplete sensory records. Across seven real-world single-arm and bimanual tasks, it raises the average success rate over concatenation-based fusion from 0.35 to 0.70 with ACT and from 0.52 to 0.80 with $\pi_{0.5}$. At a fixed complete-demonstration budget, additional demonstrations without touch or sound raise DexMuse (ACT) from 0.25 to between 0.42 and 0.48.

AI USE STATEMENT

The research questions, the DexMuse method, the robot platforms, and the experimental design were conceived by the authors. Generative AI tools (Claude and ChatGPT) were used to help implement training and analysis code, to process data and analyze results, to create and edit figures, to search for and verify related literature, and to draft and polish parts of the text. They were not used to collect or synthesize demonstrations or sensor data, since all demonstrations were teleoperated and all real-world trials were run and scored by the authors. All AI-assisted code was reviewed and tested by the authors, figures and reported numbers were checked against the underlying evaluation and analysis records, every reference was verified against its publisher or arXiv record, and all text was revised by the authors. The authors take full responsibility for the content of this work, including text, figures, and code produced with the aid of generative AI.

ETHICS STATEMENT

This work studies robot manipulation learned from demonstrations recorded in a laboratory. All demonstrations were collected by the authors through teleoperation and consist of images of the robot workspace, tactile signals, contact-microphone audio, and robot states; they involve no human-subject study and contain no personally identifiable information. All real-world experiments were conducted and supervised by the authors. We do not foresee specific negative societal impacts beyond the general considerations for autonomous manipulation systems, such as the need for safety supervision during deployment.

REPRODUCIBILITY STATEMENT

Section 3 and Appendix A describe the architecture of DexMuse, its interfaces to ACT and $\pi_{0.5}$, and their training objectives, including encoder sizes, the fingertip sampling kernel, gate initializations, and loss functions. Appendix B describes the robot platforms, sensors, and task objects; Appendix C describes the teleoperation system and the number of demonstrations per task; Appendix D lists the training hyperparameters for both backbones; and Appendix E describes the baseline implementations. Every reported success rate is measured over 13 real-world trials per method, task, and condition.

REFERENCES

Iris Andrussow, Jans Solano, Benjamin A. Richardson, Georg Martius, and Katherine J. Kuchenbecker. Adding internal audio sensing to internal vision enables human-like in-hand fabric recognition with soft robotic fingertips. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2025.

Haonan Chen, Jiaming Xu, Hongyu Chen, Kaiwen Hong, Binghao Huang, Chaoqi Liu, Jiayuan Mao, Yunzhu Li, Yilun Du, and Katherine Driggs-Campbell. Multi-modal manipulation via multi-modal policy consensus. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026a.

Ruoqu Chen, Feixiang Ruan, Liu Cao, Zihao Wang, Botian Xu, Shiqin Tong, Jiajun Liu, Mingzhi Pei, Chenyu Zhang, Wanli Xing, Kaifeng Zhang, and Mengdi Xu. DEX-X: Learning visual-tactile dexterous manipulation from human videos with simulated interaction. In *Conference on Robot Learning (CoRL)*, 2026b.

Yizhou Chen, Andrea Sipos, Mark Van der Merwe, and Nima Fazeli. Visuo-tactile transformers for manipulation. In *Conference on Robot Learning (CoRL)*, 2022.

Jaden Clark, Changhao Wang, Yihuai Gao, Seongheon Hong, Hojung Choi, Mark Cutkosky, Yifan Hou, and Shuran Song. Multisensory continual learning: Adapting pretrained visuomotor policies to force. In *Conference on Robot Learning (CoRL)*, 2026.

Maximilian Du, Olivia Y. Lee, Suraj Nair, and Chelsea Finn. Play it by ear: Learning skills amidst occlusion through audio-visual imitation learning. In *Robotics: Science and Systems (RSS)*, 2022.

Ruoxuan Feng, Di Hu, Wenke Ma, and Xuelong Li. Play to the score: Stage-guided dynamic multi-sensory fusion for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2024.

Ruoxuan Feng, Jiangyu Hu, Wenke Xia, Tianci Gao, Ao Shen, Yuhao Sun, Bin Fang, and Di Hu. AnyTouch: Learning unified static-dynamic representation across multiple visuo-tactile sensors. In *International Conference on Learning Representations (ICLR)*, 2025.

Niklas Funk, Changqi Chen, Tim Schneider, Georgia Chalvatzaki, Roberto Calandra, and Jan Peters. On the importance of tactile sensing for imitation learning: A case study on robotic match lighting. *IEEE Robotics and Automation Letters*, 11(5):6218–6225, 2026.

Irmak Guzey, Ben Evans, Soumith Chintala, and Lerrel Pinto. Dexterity from touch: Self-supervised pre-training of tactile representations with robotic play. In *Conference on Robot Learning (CoRL)*, 2023.

Liang Heng, Haoran Geng, Kaifeng Zhang, Pieter Abbeel, and Jitendra Malik. ViTacFormer: Learning cross-modal representation for visuo-tactile dexterous manipulation. In *Robotics: Science and Systems (RSS)*, 2026.

Carolina Higuera, Akash Sharma, Chaithanya Krishna Bodduluri, Taosha Fan, Patrick Lancaster, Mrinal Kalakrishnan, Michael Kaess, Byron Boots, Mike Lambeta, Tingfan Wu, and Mustafa Mukadam. Sparsh: Self-supervised touch representations for vision-based tactile sensing. In *Conference on Robot Learning (CoRL)*, 2024.

Carolina Higuera, Akash Sharma, Taosha Fan, Chaithanya Krishna Bodduluri, Byron Boots, Michael Kaess, Mike Lambeta, Tingfan Wu, Zixi Liu, Francois Robert Hogan, and Mustafa Mukadam. Tactile beyond pixels: Multisensory touch representations for robot manipulation. In *Conference on Robot Learning (CoRL)*, pp. 105–123, 2025.

Binghao Huang, Yixuan Wang, Xinyi Yang, Yiyue Luo, and Yunzhu Li. 3D-ViTac: Learning fine-grained manipulation with visuo-tactile sensing. In *Conference on Robot Learning (CoRL)*, 2024.

Guangqi Jiang, Haoran Chang, Ri-Zhao Qiu, Yutong Liang, Mazeyu Ji, Jiyue Zhu, Zhao Dong, Xueyan Zou, and Xiaolong Wang. Gsworld: Closed-loop photo-realistic simulation suite for robotic manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026a.

Guangqi Jiang, Yutong Liang, Jianglong Ye, Jia-Yang Huang, Changwei Jing, Rocky Duan, Pieter Abbeel, Xiaolong Wang, and Xueyan Zou. Cross-hand latent representation for vision-language-action models. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026b.

Changwei Jing, Jai Krishna Bandi, Jianglong Ye, Yan Duan, Pieter Abbeel, Xiaolong Wang, and Sha Yi. Contact-aware neural dynamics. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Joshua Jones, Oier Mees, Carmelo Sferrazza, Kyle Stachowicz, Pieter Abbeel, and Sergey Levine. Beyond sight: Finetuning generalist robot policies with heterogeneous sensors via language grounding. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

Rickmer Krohn, Erik Helmut, Niklas Funk, Jan Peters, Vignesh Prasad, and Georgia Chalvatzaki. Multi-resolution tactile imitation learning for contact-rich robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2026.

Yi-Lun Lee, Yi-Hsuan Tsai, Wei-Chen Chiu, and Chen-Yu Lee. Multimodal prompting with missing modalities for visual recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14943–14952, 2023.

Hao Li, Yizhi Zhang, Junzhe Zhu, Shaoxiong Wang, Michelle A Lee, Huazhe Xu, Edward Adelson, Li Fei-Fei, Ruohan Gao, and Jiajun Wu. See, hear, and feel: Smart sensory fusion for robotic manipulation. In *Conference on Robot Learning (CoRL)*, 2022.

Hong Li, Yankang Dong, Yue Xu, Yihan Tang, Mingzhu Li, Jiamin Qiu, Qihang Yao, Xing Zhu, Yujun Shen, Nan Xue, and Yong-Lu Li. Tac-DINO: Learning vision-tactile features with patch alignment. *arXiv preprint arXiv:2606.12069*, 2026a.

Xukun Li, Yu Sun, Lei Zhang, Bosheng Huang, Yibo Peng, Yuan Meng, Haojun Jiang, Shaoxuan Xie, Guocai Yao, Alois Knoll, Zhenshan Bing, Xinlong Wang, and Zhenguo Sun. DECO: Decoupled multimodal diffusion transformer for bimanual dexterous manipulation with a plugin tactile adapter. In *International Conference on Machine Learning (ICML)*, 2026b.

Zinan Li, Yiyang Ling, Yuming Gu, Binghao Huang, Chenhao Liang, Sharfin Islam, Hisham Bedri, John Chirikjian, Yunzhu Li, Stefanos Nikolaidis, and Daniel Seita. FELT: Generating tactile signals from vision for visuo-tactile manipulation. *arXiv preprint arXiv:2607.20683*, 2026c.

Toru Lin, Yu Zhang, Qiyang Li, Haozhi Qi, Brent Yi, Sergey Levine, and Jitendra Malik. Learning visuotactile skills with two multifingered hands. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

Xingyuming Liu, Ruichun Ma, Heyu Guo, Qixiu Li, Qingwen Yang, Lin Luo, Shiqi Jiang, Chenren Xu, Jiaolong Yang, and Baining Guo. MuseVLA: An adaptive multimodal sensing vision-language-action model for robotic manipulation. *arXiv preprint arXiv:2606.17598*, 2026a.

Zeyi Liu, Cheng Chi, Eric Cousineau, Naveen Kuppuswamy, Benjamin Burchfiel, and Shuran Song. ManiWAV: Learning robot manipulation from in-the-wild audio-visual data. In *Conference on Robot Learning (CoRL)*, 2024.

Zhuoyang Liu, Jiaming Liu, Jiadong Xu, Nuowei Han, Chenyang Gu, Hao Chen, Kaichen Zhou, Renrui Zhang, Kai Chin Hsieh, Kun Wu, Zhengping Che, Jian Tang, and Shanghang Zhang. MLA: A multisensory language-action model for multimodal understanding and forecasting in robotic manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026b.

Shengcheng Luo, Kefei Wu, Xiaoying Zhou, Wanlin Li, Ziyuan Jiao, and Chenxi Xiao. RGB-S: Image-aligned tactile saliency for robust dexterous manipulation. *arXiv preprint arXiv:2606.08765*, 2026.

Mengmeng Ma, Jian Ren, Long Zhao, Sergey Tulyakov, Cathy Wu, and Xi Peng. SMIL: Multimodal learning with severely missing modality. In *AAAI Conference on Artificial Intelligence*, pp. 2302–2310, 2021.

Mengmeng Ma, Jian Ren, Long Zhao, Davide Testuggine, and Xi Peng. Are multimodal transformers robust to missing modality? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18177–18186, 2022.

Yuemin Mao, Uksang Yoo, Jean Oh, Jonathan Francis, and Jeffrey Ichnowski. VibeAct: Vibration to actions for contact-rich reactive robot dexterity. *arXiv preprint arXiv:2606.27344*, 2026.

Jared Mejia, Victoria Dean, Tess Hellebrekers, and Abhinav Gupta. Hearing touch: Audio-visual pretraining for contact-rich manipulation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Ken Nakahara and Roberto Calandra. Learning gentle grasping using vision, sound, and touch. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

Natalia Neverova, Christian Wolf, Graham Taylor, and Florian Nebout. ModDrop: Adaptive multimodal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38 (8):1692–1706, 2016.

Chang Nie, Tianchen Deng, Guangming Wang, Zhe Liu, and Hesheng Wang. Towards the vision-sound-language-action paradigm: The HEAR framework for sound-centric manipulation. *The International Journal of Robotics Research*, 2026.

Dantong Niu, Zhuoyang Liu, Zekai Wang, Boning Shao, Zhao-Heng Yin, Anirudh Pai, Yuvan Sharma, Stefano Saravalle, Ruijie Zheng, Jing Wang, Ryan Punamiya, Mengda Xu, Yuqi Xie, Yunfan Jiang, Letian Fu, Konstantinos Kallidromitis, Matteo Gioia, Junyi Zhang, Jiaxin Ge, Haiwen Feng, Fabio Galasso, Wei Zhan, David M. Chan, Yutong Bai, Roei Herzig, Jiahui Lei, Li Fei-Fei, Ken Goldberg, Jitendra Malik, Pieter Abbeel, Yuke Zhu, Danfei Xu, Linxi Fan, and Trevor Darrell. T-Rex: Tactile-reactive dexterous manipulation. In *Conference on Robot Learning (CoRL)*, 2026.

Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, Jianlan Luo, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Robotics: Science and Systems (RSS)*, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *Conference on Robot Learning (CoRL)*, pp. 17–40, 2025.

Carmelo Sferrazza, Younggyo Seo, Hao Liu, Youngwoon Lee, and Pieter Abbeel. The power of the senses: Generalizable manipulation from vision and touch through masked multimodal learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024.

Abitha Thankaraj and Lerrel Pinto. That sounds right: Auditory self-supervision for dynamic robot manipulation. In *Conference on Robot Learning (CoRL)*, 2022.

Mark Van der Merwe, Mohamad Louai Shehab, Jayjun Lee, Youngsun Wi, Yinpei Dai, Dmitry Berenson, and Nima Fazeli. TAP-VLA: Tactile annotation prompting for vision language action models. *arXiv preprint arXiv:2606.29089*, 2026.

Lirui Wang, Xinlei Chen, Jialiang Zhao, and Kaiming He. Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Renhao Wang, Haoran Geng, Tingle Li, Feishi Wang, Gopala Anumanchipalli, Trevor Darrell, Boyi Li, Pieter Abbeel, Jitendra Malik, and Alexei A Efros. The sound of simulation: Learning multimodal sim-to-real robot policies with generative audio. In *Conference on Robot Learning (CoRL)*, 2025.

Zhuowei Xu, Zilin Si, Oliver Kroemer, and Zeynep Temel. A multi-modal tactile fingertip design for robotic hands to enhance dexterous manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026.

Han Xue, Jieji Ren, Wendi Chen, Gu Zhang, Yuan Fang, Guoying Gu, Huazhe Xu, and Cewu Lu. Reactive diffusion policy: Slow-fast visual-tactile policy learning for contact-rich manipulation. In *Robotics: Science and Systems (RSS)*, 2025.

Fengyu Yang, Chao Feng, Ziyang Chen, Hyoungseob Park, Daniel Wang, Yiming Dou, Ziyao Zeng, Xien Chen, Rit Gangopadhyay, Andrew Owens, and Alex Wong. Binding touch to everything: Learning unified multimodal tactile representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Jianglong Ye, Lai Wei, Guangqi Jiang, Changwei Jing, Xueyan Zou, and Xiaolong Wang. From power to precision: Learning fine-grained dexterity for multi-fingered robotic hands. *arXiv preprint arXiv:2511.13710*, 2025.

Kelin Yu, Yunhai Han, Qixian Wang, Vaibhav Saxena, Danfei Xu, and Ye Zhao. MimicTouch: Leveraging multi-modal human tactile demonstrations for contact-rich manipulation. In *Conference on Robot Learning (CoRL)*, 2024.

Kelin Yu, Haode Zhang, Harish Ravichandar, Yunhai Han, and Ruohan Gao. OmniTacTune: Policy-agnostic real-world RL for tactile residual adaptation of visual policies. In *Conference on Robot Learning (CoRL)*, 2026.

Haoran Yuan, Zekai Wang, Boning Shao, Haoran Lu, Trevor Darrell, Ismini Lourentzou, and Wei Zhan. DexTacWAM: A visuo-tactile world-action model for dexterous manipulation. *arXiv preprint arXiv:2609.24976*, 2026.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Chi Zhang, Penglin Cai, Haoqi Yuan, Chaoyi Xu, and Zongqing Lu. UniTacHand: Unified spatio-tactile representation for human to robotic hand skill transfer. *arXiv preprint arXiv:2512.21233*, 2025.

Kaidi Zhang, Heng Zhang, Zhengtong Xu, Zhiyuan Zhang, Md Rakibul Islam Prince, Xiang Li, Xiaojing Han, Yuhao Zhou, Arash Ajoudani, and Yu She. TacVLA: Contact-aware tactile fusion for robust vision-language-action manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2026a.

Xuancheng Zhang, Xuetao Liu, Qianying Tang, Jizhe Wang, Runsheng Wang, Zhijing Cheng, Bochen Lin, Haoran Wen, Ming Li, Kun Zhan, and Yu Liu. ME-Dex 1.0: Bringing heterogeneous tactile sensing into world action modeling. *arXiv preprint arXiv:2609.21449*, 2026b.

Zhemeng Zhang, Jiahua Ma, Xincheng Yang, Xin Wen, Yuzhi Zhang, Boyan Li, Yiran Qin, Jin Liu, Can Zhao, Li Kang, Haoqin Hong, Zhenfei Yin, Philip Torr, Hao Su, Ruimao Zhang, and Daolin Ma. TouchGuide: Inference-time steering of visuomotor policies via touch guidance. In *Robotics: Science and Systems (RSS)*, 2026c.

Jialiang Zhao, Naveen Kuppuswamy, Siyuan Feng, Benjamin Burchfiel, and Edward Adelson. PolyTouch: A robust multi-modal tactile sensor for contact-rich manipulation using tactile-diffusion policies. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. In *Robotics: Science and Systems (RSS)*, 2023.

Xinyue Zhu, Binghao Huang, and Yunzhu Li. Touch in the Wild: Learning fine-grained manipulation with a portable visuo-tactile gripper. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

A IMPLEMENTATION OF HAND-GROUNDED FUSION

A.1 REGION AND AUDIO ENCODERS

All interaction features have dimension $d = 256$. Multi-head acoustic attention uses eight heads and $L = 4$ temporal audio tokens per active hand. The same per-hand module is applied to the active hands of a task. Five tokens represent the fingers of XHand; six tokens represent the five fingers and palm of XHand Pro. The module therefore produces five or six tokens for a single hand, and ten or twelve for two hands with the corresponding layout.

Tactile encoding. For dense XHand observations, the sensor map is partitioned into five contiguous finger regions. Each single-channel region is processed by three 3×3 convolutions with channels $(16, 32, 64)$ and strides $(1, 2, 2)$. Each convolution is followed by GroupNorm with $(4, 8, 8)$ groups and GELU. Global average pooling and a linear projection produce one d-dimensional token per finger. XHand Pro provides 18 sensors with four recorded channels per sensor. A shared MLP with widths $(4, d, d)$ and GELU embeds each sensor, and a learned sensor identity is added. The three sensors assigned to each region are averaged and layer-normalized to produce its tactile token.

Proprioception and identity. Separate two-layer MLPs encode the seven-dimensional arm state and the joints associated with each region. A learned region embedding is added to their sum before layer normalization. XHand uses three joints for the thumb and index finger and two for each remaining finger, with the shorter vectors zero-padded. XHand Pro uses four joints per finger and one palm joint, with the palm vector padded to length four. The same learned region identities are reused across active hands. In bimanual ACT, fixed left/right encodings, $(-1, 0, \ldots, 0)$ and $(1, 0, \ldots, 0)$, are added to the region positional embeddings. In $\pi_{0.5}$ they are added to interaction features before final normalization and projection to the prefix dimension.

Acoustic encoding. Audio is resampled to 16 kHz. The input waveform concatenates 120 consecutive blocks of 800 samples, covering six seconds up to the stream-corrected observation timestamp. Unavailable history at the start of an episode is zero-padded. For waveform sample s_i, pre-emphasis computes $e_i = s_i - 0.97s_{i-1}$, with $e_0 = s_0$. Signed logarithmic compression gives $\hat{s}_i = \operatorname{sign}(e_i)\log(1 + 64|e_i|)$. Three temporal convolutions use channels $(16, 32, 64)$, kernel sizes $(15, 15, 9)$, and stride four at each layer. They are followed by GroupNorm with $(4, 8, 8)$ groups and GELU. Temporal average pooling produces four ordered features, which are linearly projected to d dimensions and combined with learned temporal-position embeddings.

A.2 PROJECTION AND LOCAL FEATURE EXTRACTION

Camera intrinsics and the rigid transform from the relevant robot base to the camera optical frame define the fingertip projection. Forward kinematics uses physical joint angles aligned to image timestamps; the policy's proprioceptive encoders use normalized state vectors. Projection validity requires positive depth and an image coordinate within the view. Local feature extraction uses the image coordinates after the resizing and padding applied to that view and, for $\pi_{0.5}$, after its geometric training augmentation.

The neighborhood offsets are $\{-r, 0, r\} \times \{-r, 0, r\}$, with $r = 7$ pixels in the corresponding backbone image coordinate system. The unnormalized pooling kernel is the outer product of $(1, 2, 1)$ with itself. Invalid samples receive zero weight; normalization uses $\epsilon = 10^{-6}$, and a fully invalid neighborhood returns a zero feature. ACT converts image coordinates to the normalized grid convention used by bilinear sampling with `align_corners=false`. For a width W, its horizontal coordinate is $2(u + 0.5)/W - 1$, with the analogous expression for height. The $\pi_{0.5}$ adapter expresses locations in its native patch-grid coordinates and scales the offsets by the 14-pixel patch size. Image transformations and fingertip coordinates are transformed together.

The six-dimensional geometry input concatenates the two normalized sampling coordinates, three camera-frame coordinates, and projection-validity flag. Its MLP has widths $(6, d, d)$ with GELU. The sampled image feature is projected to d dimensions and added to the geometry embedding. The visual residual applies GELU and a zero-initialized $d \times d$ output projection. The visual and acoustic gates have zero-initialized weights and initial bias -2. During training, the visual and acoustic

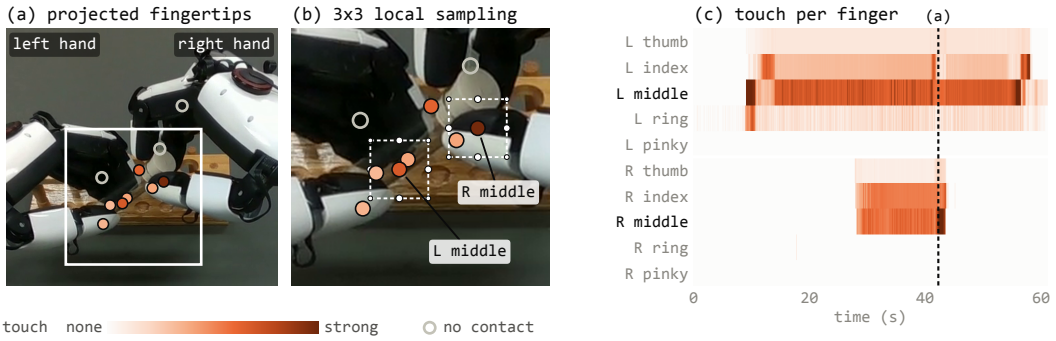


Figure 8: **Calibrated fingertip projection on an Insert Stopper demonstration.** (a) While the stopper is inserted, forward kinematics and camera calibration place each of the ten fingertips in the image; fill color gives the finger's current touch, and hollow rings mark fingers without contact. (b) Enlarged region of (a) with the 3×3 sampling neighborhoods of the left and right middle fingers ($r = 7$ pixels at the ACT input resolution; dot size follows the pooling weights). (c) Touch of each finger over the demonstration, measured by the policy's tactile input relative to the start of the demonstration; the dashed line marks the frame in (a).

residuals of each hand are independently dropped with probability 0.1. These residual masks are separate from demonstration-level sensor-availability masks.

Figure 8 shows the projected fingertips, the sampling neighborhoods, and the per-finger touch on an Insert Stopper demonstration.

A.3 Backbone interfaces and objectives

ACT architecture. The visual backbone is frozen DINOv2 ViT-S/14 (Oquab et al., 2024). Spatial features support both fingertip sampling and global scene tokens. The interaction sequence contains the ACT latent token, a projected robot-state token, and hand interaction tokens. The scene sequence contains projected global visual tokens. A shared four-layer Transformer encoder processes the two sequences independently. The seven-layer action decoder uses 64 learned action queries; each layer applies self-attention, cross-attention to the interaction memory, gated cross-attention to the scene memory, and a feed-forward network. As in the original ACT implementation, the action head reads the output of the first decoder layer, in both DexMuse (ACT) and MM-ACT. The decoder width is 256, the feed-forward width is 3200, and attention uses eight heads. The global-scene gate is conditioned on the current query and scene update, with zero initial weights and bias -2.

Let $q_\phi(z \mid \mathbf{a}_t, q_t)$ be ACT's recognition posterior, $p(z) = \mathcal{N}(0, I)$ its prior, and $\mathcal{V}_t$ the non-padded action indices. Training minimizes

$$\ell_{\mathrm{ACT}} = \mathbb{E}_{z \sim q_\phi}\left[\frac{1}{|\mathcal{V}_t| d_a} \sum_{j \in \mathcal{V}_t} \|\hat{a}_{t,j}(z) - a_{t,j}\|_1\right] + \beta\, D_{\mathrm{KL}}(q_\phi \| p), \tag{6}$$

where d_a is the action dimension and $\beta = 10$. The recognition encoder has four layers, produces a 32-dimensional latent, and consumes robot state and the demonstrated action chunk. At deployment, $z = 0$. The action representation and horizon remain those of the ACT policy.

$\pi_{0.5}$ architecture. The VLA retains its native SigLIP image encoder (Zhai et al., 2023), language tokens, and flow-matching action expert. The local sampler uses its spatial image features before token flattening. Interaction tokens are projected to the pretrained language-model width and appended after the image tokens and before the language tokens in the observation prefix. Their prefix attention mask allows joint observation conditioning, and action suffix tokens attend to this enriched prefix. The action horizon is 64.

For a normalized action chunk $\mathbf{a}$, draw $\epsilon \sim \mathcal{N}(0, I)$ and noise level $s = 0.999b + 0.001$, where $b \sim \mathrm{Beta}(1.5, 1)$. The interpolation $x_s = (1 - s)\mathbf{a} + s\epsilon$ has target velocity $\epsilon - \mathbf{a}$. Let $\mathcal{V}$ denote the non-padded action timesteps. The loss is

$$\ell_{\pi_{0.5}} = \mathbb{E}_{\epsilon,s}\left[\frac{1}{|\mathcal{V}|d_a}\sum_{j\in\mathcal{V}}\|v_\theta(x_s, s; \mathrm{prefix})_j - (\epsilon_j - \mathbf{a}_j)\|_2^2\right]. \tag{7}$$

Action sampling starts from Gaussian noise at $s = 1$ and integrates the predicted velocity toward $s = 0$ using the policy's flow sampler. Sensor fusion changes the conditioning prefix while preserving this training and sampling formulation.

Mixed-data sampling. The ACT episode-grouped sampler shuffles episode order and the action windows within each episode, then forms minibatches from each episode's windows, retaining the final partial batch. A traversal visits all windows. Complete and incomplete records use the same action objective, with their relative exposure determined by the resulting batches rather than a prescribed mixture weight.

A.4 Configurations for feature analysis

The feature analysis in Section 4.3 compares two ACT policies, MM-ACT and DexMuse (ACT), which receive the same vision, touch, sound, and robot-state inputs. MM-ACT concatenates per-modality tokens, while DexMuse supplies hand-region tokens alongside global image tokens through the interface described in Appendix A.3.

B Real-world environment

Figure 9 shows the two real-world platforms and the task objects. Both platforms use UFACTORY xArm7 arms, which are 7-DoF manipulators, and an external Intel RealSense D435i camera that is mounted on a stand, provides the RGB observations, and is calibrated to the robot base (Appendix A.2).

Single-arm platform. One xArm7 carries an XHand Pro and performs the three single-arm tasks (Flip Bagel, Erase Board, and Sort Bank). Its joint and tactile layouts are described in Appendix A.1.

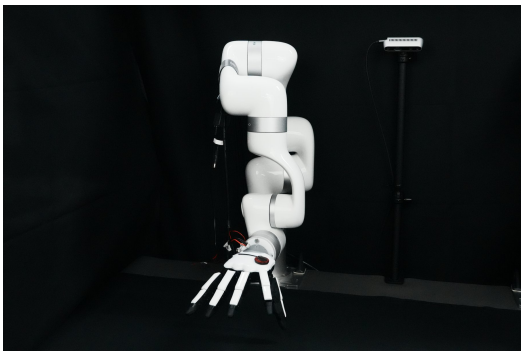
Bimanual platform. Two xArm7 arms are mounted side by side, each carrying an XHand, and perform the four bimanual tasks (Insert Stopper, Pack Candies, Insert Knife, and Pour Candies). The XHand layouts are described in Appendix A.1.

Audio and synchronization. A contact microphone on the back of each hand records interaction sounds, which are aligned with the camera and robot-state streams (Appendix A.1).

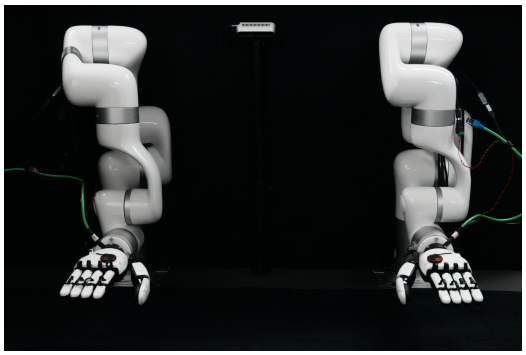
Workspace and objects. The tasks are performed on a padded work surface in front of the arms (Figure 9c). The objects are a frying pan, a spatula, and a toy bagel (Flip Bagel); a whiteboard and an eraser (Erase Board); an opaque piggy bank and two printed placement targets (Sort Bank); a glass tube, a cork stopper, and a wooden tube rack (Insert Stopper); a paper bag and candies (Pack Candies); a toy knife and its holder (Insert Knife); and a mesh bag of candies with a transparent container (Pour Candies).

C Real-world data collection

We collect demonstrations with a teleoperation system that tracks the operator's wrists and hands. A Meta Quest 3 headset provides the wrist poses, and a Wuji motion-capture glove provides the hand poses. Each wrist pose is converted to arm joint positions by inverse kinematics, and each hand pose is retargeted to the joint positions of the dexterous hand; in bimanual tasks, the two arms follow the operator's two wrists and hands. Table 5 lists the number of demonstrations collected for each task.



(a) Single-arm platform



(b) Bimanual platform



(c) Task objects

Figure 9: **Real-world platforms and objects.** (a) An xArm7 with an XHand Pro and an external RealSense D435i camera. (b) Two xArm7 arms with XHand hands and a camera between them. A contact microphone is mounted on the back of each hand. (c) Objects used in the seven tasks.

	Single-arm			**Bimanual**			
	Flip Bagel	**Erase Board**	**Sort Bank**	**Insert Stopper**	**Pack Candies**	**Insert Knife**	**Pour Candies**
Demonstrations	74	125	74	55	61	60	58

Table 5: Number of teleoperated demonstrations per task.

D DexMuse training details

This section describes how DexMuse is trained; its architectures are given in Appendix A. Within each backbone, a single set of hyperparameters is used for all seven tasks, and all policies are trained on NVIDIA RTX PRO 5000 Blackwell GPUs.

ACT. We train DexMuse (ACT) on a single GPU with AdamW for 40,000 steps at an effective batch size of 256, using a learning rate of 10^{-4} with 500 warm-up steps and cosine decay to 10^{-5}, weight decay 10^{-4}, and gradient clipping at 1.0. The DINOv2 image encoder is frozen, and the policy predicts 64-step action chunks from 224×398 images with a KL weight of $\beta = 10$. Every data condition in Table 4, including mixed training with incomplete demonstrations, uses the same 40,000-step budget, so gains from additional demonstrations do not come from longer training.

$\pi_{0.5}$. We initialize DexMuse ($\pi_{0.5}$) from the official open-source $\pi_{0.5}$ base checkpoint, with a Gemma 2B vision-language backbone and a Gemma 300M action expert, and fine-tune all parameters jointly with the flow-matching objective of Appendix A.3. Training uses AdamW for 40,000 steps at a batch size of 36 on two GPUs with fully sharded data parallelism, a peak learning rate of 2×10^{-5} with 2,000 warm-up steps and cosine decay to 2×10^{-6}, and gradient clipping at 1.0. Images are resized with padding to 224×224, and the action horizon is 64. Following OpenPI, training images are augmented with random crops, small rotations, and color jitter, and the same geometric transform is applied to the projected fingertip coordinates.

Regularization. For both backbones, the visual and acoustic residuals of each hand are dropped with probability 0.1 during training (Appendix A.2).

E Baseline implementation details

All baselines are trained on the same demonstrations as DexMuse for each task. They predict the same arm-and-hand action chunks and receive the same robot state.

Multimodal backbones. MM-ACT and MM-$\pi_{0.5}$ are the concatenation counterparts of DexMuse within each policy family and serve as matched baselines for its fusion design. They receive the same RGB, tactile, audio, and robot-state inputs and use the same image encoder as DexMuse. Each sensory stream is encoded into its own tokens, which are concatenated with the image tokens and passed to the backbone: as encoder memory for ACT, and as additional observation-prefix tokens for $\pi_{0.5}$. No fingertip projection, regional grouping, or cross-modal attention is applied, so the two variants differ from DexMuse in how the modalities are organized and fused, including the backbone interface described in Section 3.2, rather than in the information they receive.

Missing modalities in MM-ACT. For the mixed-data experiments in Table 4, MM-ACT uses the same fixed episode-level availability masks, the same handling of missing touch and sound, and the same modality dropout as DexMuse: an absent tactile or audio stream is masked rather than replaced, so it contributes no evidence, and complete and incomplete demonstrations optimize the same ACT action objective. The two designs therefore differ only in how the available modalities are fused.

Published methods. ManiWAV, Touch in the Wild, Sparsh-X, and FuSe are implemented from the official code released by their authors. We keep their architectures and training objectives and replace only the sensor, embodiment, and action interfaces: our contact-microphone audio, tactile arrays, and scene camera replace the original sensors, and the policy heads output our arm-and-hand actions. Each method receives the modalities its design supports: images and contact audio for ManiWAV, images and touch for Touch in the Wild, the multisensory tactile representation of Sparsh-X together with scene images, and images, touch, and audio for FuSe.

F Extended related work

Multisensory sensing and fusion. Sound has supported self-supervised interaction learning (Thankaraj & Pinto, 2022), audio-visual transfer to contact microphones (Mejia et al., 2024), streaming acoustic memory (Nie et al., 2026), grasp supervision (Nakahara & Calandra, 2025), and contact and slip estimation (Mao et al., 2026), and has been combined with touch in fingertip sensors (Xu et al., 2026; Andrussow et al., 2025) and multisensory tactile representations (Yang et al., 2024; Higuera et al., 2025). Policies are adapted or augmented with new sensors through pretrained tactile representations (Higuera et al., 2024; Feng et al., 2025), language grounding (Jones et al., 2025), tactile adapters (Li et al., 2026b), inference-time guidance (Zhang et al., 2026c), tactile residuals (Yu et al., 2026), or policy consensus (Chen et al., 2026a), and vision–touch policies use selective tactile tokens (Zhang et al., 2026a), multi-rate processing (Xue et al., 2025; Niu et al., 2026), or contrastive pairs from projected gripper locations (Liu et al., 2026b). None of these anchors both touch and sound at calibrated fingertip locations in the image, which DexMuse uses to organize fusion around the hand.

Heterogeneous and incomplete data. Multisensory supervision can be expanded by collecting visuotactile or human demonstrations (Zhu et al., 2025; Yu et al., 2024; Zhang et al., 2025) or by synthesizing contact, tactile, or sensor signals (Chen et al., 2026b; Zhang et al., 2026b; Li et al., 2026c; Liu et al., 2026a). Missing inputs are handled by masking or dropping observations (Sferrazza et al., 2024; Funk et al., 2026; Krohn et al., 2026; Octo Model Team et al., 2024; Neverova et al., 2016), mapping heterogeneous sensors into a shared Transformer (Wang et al., 2024), reconstructing features (Ma et al., 2021), robust fusion (Ma et al., 2022), missing-modality prompts (Lee et al., 2023), or learned mask tokens, as in the multisensory continual learning of MuSe (Clark et al., 2026). These methods treat a modality as a whole stream; availability-aware mixed training instead masks missing touch or sound within each hand region while its fingertip anchor remains (Eq. 5).