Adaptive Modality Selection in Multimodal Wearable and XR Interfaces

Introduction

Multimodal user interfaces allow people to interact via **gesture**, **voice**, **brain signals (EEG)**, **touch/click**, and more. This is especially powerful in wearables and XR (AR/VR) systems, where natural inputs like hand gestures, speech commands, and even neural cues offer intuitive control. However, each modality has limitations – for example, gaze-based controls can suffer the "Midas Touch" problem and accuracy issues [1], gesture input can cause arm fatigue over time [2], and voice commands can increase cognitive load or be impractical in noisy/public settings [2]. No single input mode is optimal in all contexts. This motivates **adaptive multimodal systems** that can dynamically select or blend input modalities based on context. The challenge is deciding **when and how to switch** between gesture, voice, EEG, or click/touch input to maximize usability.

Recent research explores **machine learning approaches**, including *contextual multi-armed bandits* and *reinforcement learning (RL)*, to optimize modality selection over time. These approaches use real-time context (user state, environment, input success history, etc.) to **"learn" the best input mode for each situation**, rather than relying solely on fixed rules. The goals are to improve reliability and efficiency (by using the modality likely to succeed), to **balance exploration vs. exploitation** (occasionally try alternative inputs to learn if they work better), and to maintain a seamless, non-intrusive user experience. In this report, we survey academic and prototype systems employing such techniques, and also discuss practical strategies (including heuristic arbitration and UI design patterns) used in both research and commercial multimodal interfaces.

Contextual Factors for Input Mode Decision-Making

Adaptive systems leverage a variety of **contextual factors** to decide which modality to use at any moment. Important context cues include:

- **User pose and device usage zone:** e.g. whether the user's hands are visible to cameras or busy with other tasks (*pose zone*), or whether the user is looking at a target (for gaze interaction). If hands are out of view or occupied, a smart system might favor voice or EEG input over gesture.
- **Environmental conditions:** e.g. ambient noise level, lighting, or social setting. In a loud environment, speech recognition accuracy drops, so the system can switch to silent inputs or tactile clicks. Conversely, in quiet private settings voice may be very convenient. Public settings also raise social comfort issues – large hand motions or speaking aloud might feel awkward [3]. Systems account for this by favoring more subtle modalities (like micro-gestures or mental commands) when social context demands discretion [3] [4].
- **Recent input success/failure and confidence:** The system monitors how well each modality has been performing. For example, if a user's last few voice commands were misrecognized (low success rate), an adaptive interface might suggest an alternative (e.g. gesture or a manual click) for the next

action. Likewise, if a hand gesture was attempted but not recognized (perhaps due to occlusion), the system can promptly fall back to another mode. Modern approaches often use **confidence scores** from recognizers (speech confidence, gesture classification probability, etc.) as contextual features for arbitration.

- **User preferences or physiological state:** Some systems also model the user's cognitive load, fatigue, or preference. For instance, a neuroadaptive interface could detect high cognitive workload via EEG and then avoid modes that require intense concentration, instead choosing a simpler input method. Papakostas et al. (2025) describe a wearable architecture that combines behavioral inputs with lightweight EEG and other sensors to infer user workload and **adapt the interface dynamically** (e.g. adjusting feedback modality or assistance level) to optimize user experience [5] [6]. This underscores how internal context (user state) can guide modality selection in addition to external context.

By combining such context signals, an adaptive system forms a *state representation* that informs which input channel is likely to be most effective at each moment. The next sections discuss how different strategies use this context: from simple heuristics to advanced learning algorithms.

Modality Arbitration Strategies in Multimodal Systems

When multiple input modalities are available, the system needs an **arbitration mechanism** to decide which inputs to accept or prompt. Approaches range from fixed heuristic rules to AI-driven fusion:

- **Rule-Based Fallback Heuristics:** Many practical systems use predetermined rules and fallback sequences. For example, a wearable AR headset might be programmed to *always listen* for voice commands, but if the microphone input confidence is low (e.g. due to noise) it then waits for a hand gesture or a click input as a backup. Similarly, if an error is detected with gaze pointing on an AR device, the system could automatically switch to an alternate pointer (like head-controlled cursor or a handheld controller) [7]. Sidenmark et al. implement an **error-aware gaze interface** that *automatically* transitions to a head-pointing modality when eye-tracking error exceeds a threshold, and even continuously blends the two modalities with weightings based on the error ratio [7]. These heuristics improve robustness by predefining which modality "takes over" under certain conditions (e.g., noise -> no voice; tracking loss -> use hardware clicker). While effective, rule-based arbitration is limited to scenarios anticipated by designers.

- **Confidence-Based Fusion and Voting:** A related strategy is to process inputs from multiple modalities in parallel and pick the result with the highest confidence or agreement. Early multimodal systems often used **recognizer confidence scores** to weight each modality's contribution. For instance, a classic voice+gesture system might parse a voice command and a simultaneous pointing gesture – if the speech recognizer is uncertain, the system could rely more on the gesture (or vice versa). Modern examples include the "Weighted Pointer" technique, where gaze and head inputs are combined: when gaze tracking is accurate, it's used for fine targeting, but as accuracy degrades, the system smoothly shifts weight to more reliable head-controlled pointing [7]. This soft arbitration avoids a hard switch; instead, both modalities contribute proportionally to the final input, ensuring a seamless handoff. Confidence-based approaches improve resilience but still require careful tuning, and they don't *learn* new rules – they apply predefined weighting logic.

- **User-Controlled Modality Switching:** It's worth noting that many commercial devices simply offer multiple input options and let the **user** choose (this is a form of manual arbitration). For example, Microsoft HoloLens 2 allows hand gesture interaction, voice commands ("select", "go home"), and a clicker or gamepad. The system does not automatically decide which you must use – the user picks what they prefer in context. Apple's 2024 Vision Pro headset similarly supports *gaze* + *pinch* gestures as the primary input, with voice available for dictation or commands, and even a fallback of connecting a keyboard. The *user* can naturally switch: e.g. if arms are tired from mid-air pinching, they might start using voice. The downside is that the burden is on the user to recognize when a modality isn't working well and to manually switch modes. Research aims to reduce this burden by making the interface itself more proactive and adaptive.

- **Hybrid Arbitration:** Some systems combine approaches – e.g. mostly user-driven but with subtle guidance from the UI. A subtle cue might be an icon or prompt appearing when the system detects a mode might be better ("Background noise high – try hand gesture" as a gentle suggestion, or highlighting the microphone icon when it detects the user saying "Hey…"). These UI cues must be carefully designed not to be intrusive. In practice, **ambient or low-attention signals** can indicate available modality switches. For instance, an AR interface could use a slight visual highlight on an affordance (like a hand icon) when it "expects" a gesture input, providing an implicit hint. Academic concepts like *Meta-UI* overlays for AR have explored showing system state and possible actions without disrupting the primary task [8] [9]. The emphasis is on keeping the user in control while *assisting* them with awareness of input options.

In summary, traditional arbitration techniques improve reliability but are largely static. **Next, we turn to learning-based approaches** that can adapt arbitration policies over time, using bandit algorithms and reinforcement learning to achieve an optimal balance between modalities.

Contextual Bandit Algorithms for Input Mode Selection

Contextual **multi-armed bandits** are a class of algorithms well-suited for iterative decision problems where each choice yields feedback. In this case, each "arm" of the bandit can be seen as choosing a particular input modality (gesture, voice, EEG, etc.), and the system receives a **reward** based on how successful that choice was (e.g. was the command executed correctly and easily?). Importantly, bandit algorithms consider the **exploration–exploitation trade-off** [10]: they will try different options to gather information (exploration), but over time favor the options that perform best (exploitation). A *contextual* bandit incorporates side information (the context) when making decisions [11]. In our scenario, the context could include all the factors discussed (pose, environment, recent performance, etc.), enabling the algorithm to learn, for example, "in noisy outdoors environments, gesture input has higher success" whereas "in quiet conditions, voice works reliably" – and many such nuanced rules – without being explicitly programmed.

How it works: At each interaction or time step, the system observes the current context state (for example: noise_level=high, hands_detected=false, prior_voice_failures=2, etc.) and must choose an input modality. A bandit algorithm like *LinUCB* (Linear Upper Confidence Bound) will compute an estimated reward for each modality based on the context features and its learned parameters, plus an uncertainty bonus for less-tried options [12]. It then picks the modality with the highest upper confidence bound, effectively balancing picking the historically best modality vs. trying alternatives that might prove better [10] [12]. Once the user attempts input via that modality, the bandit receives a reward signal – e.g., **1** if the input was successfully recognized and the task completed, or **0** if it failed and had to be corrected. Over many such trials, it updates its estimates to personalize and optimize the interface's decisions.

This approach has been demonstrated in related HCI scenarios. Lin et al. (2025) use a **contextual bandit for on-line personalization of a hand gesture recognizer**: their system treats each gesture classification as a bandit arm and uses context (features from an EMG armband's neural network) to decide which gesture label to trust, learning from binary reward feedback [12]. Effectively, the bandit learns a user-specific bias – which gestures are harder for a given user – and adjusts the model accordingly in real time. They report that users' error rates dropped significantly over multiple rounds of interaction, and even some users who initially could not complete tasks with the default gesture model succeeded once the **contextual MAB** personalization kicked in [13]. Notably, this was done without an explicit calibration step – the bandit learned from implicit success/failure signals during regular use.

Contextual bandits have also been widely applied in brain-computer interface (BCI) research, where systems must adapt to non-stationary neural signals. For example, a BCI that moves a cursor via EEG can use bandit algorithms to choose the best control policy for each user on the fly [14]. In essence, the bandit framework provides a **mathematically principled way to do modality recommendation**, continuously improving as it gathers more data. It inherently handles exploration: e.g., if voice commands have never been tried in a certain context, the algorithm will eventually give them a chance (especially if the current default is working suboptimally), which might reveal a better solution or confirm the current best.

One challenge is defining the reward in a multimodal UI. A simple approach is to use binary success/failure of the user's intended action, as in the gesture example [15]. More fine-grained reward schemes can include factors like speed (e.g. time to complete the action) or user satisfaction (perhaps inferred from error corrections or even biometric signals). Care must be taken to get reliable feedback; some systems may solicit the user's rating of an interaction to close the loop. Lu et al. (2024) and Lingler et al. (2024) showed that even using task completion as an implicit reward signal can guide an interface toward better performance without hurting user experience [16]. The key is that the adaptation happens *in the background*, gradually, and ideally the user just perceives that "the system is getting easier to use over time."

Example application: Imagine a web-based XR app that can be controlled by voice or a handheld clicker. Initially, it might not know which the user prefers or which is more effective. A contextual bandit could start by occasionally prompting the user "You can say 'Next' to navigate" while also allowing the click. If it observes that voice attempts often fail (low reward) in the current context (maybe the user's microphone quality is poor), it will lean more on click input. But if the user moves to a different environment or gets a better mic (context change), the bandit might slowly explore voice again and adapt. This continuous learning is more flexible than any static policy.

Reinforcement Learning for Modality Adaptation

While bandit algorithms decide each action independently (with context), more general **reinforcement learning** can consider longer-term sequences and more complex state representations. In an RL formulation, the *state* could encapsulate the full history or any aspects of the interaction, and a *policy* is learned to map states to modality choices in a way that maximizes cumulative reward over time. Unlike contextual bandits (which do no lookahead and only get feedback on the chosen arm), a full RL agent (e.g. using Q-learning or policy gradients) could, for example, learn a policy like "if the user has had two consecutive voice failures, switch to gesture for the next two interactions to let frustration cool down, then maybe try voice again later.” This kind of temporal strategy emerges if it improves long-term rewards (e.g., user successfully completes more tasks).

Researchers have begun exploring deep reinforcement learning in the context of **adaptive multimodal interfaces**. Carrow (2025) proposes a *deep RL-based framework for VR/AR interfaces* that can dynamically adjust input modalities, interface layout, and feedback based on the user’s behavior [17]. In their prototype, the system took in gesture, speech, and eye-tracking inputs along with performance metrics, and the RL agent learned an optimal way to fuse and switch modalities to minimize task completion time and errors [17] [18]. Experiments in simulated VR tasks showed the RL-driven adaptive UI outperformed a static interface, improving both objective metrics and subjective user satisfaction [19]. This indicates an RL policy can discover effective adaptation strategies that might be non-obvious to human designers. For example, the agent might learn to present certain interface elements only when it predicts the user is engaged (based on context), or to temporarily disable voice input if it predicts from context that a misunderstanding would occur, thus preventing an error. Such a policy is **learned from data/experience** rather than hardcoded.

Another example is the **Adaptive Multimodal Assistant (AMMA)** system (Ghamandi et al., 2024), which uses automated state tracking and planning to guide multimodal assistance in VR [20]. While not purely a deep RL approach, it highlights the trend of using AI to drive modality selection in a principled way. Similarly, neuroadaptive systems like NAMI (described earlier) can be seen as employing a form of policy: they adjusted help modality and frequency based on estimated cognitive state [6]. If such a system were extended, it could be trained (via reinforcement signals) to optimize when to use, say, an audible alert vs. a visual hint depending on what keeps the user most engaged and successful.

One interesting direction is combining **planning with exploration**. A fully autonomous RL agent in a user interface must be careful – random exploratory actions (e.g., suddenly switching to an unfamiliar EEG input modality) could confuse or annoy users. Thus, researchers often constrain the action space or use **simulated users** during training. Carrow’s work, for instance, trained the agent in simulation to avoid harming real user experience [21] [18]. Other approaches use *human-in-the-loop* training where the system explores with real users but at a controlled pace. The concept of *reinforcement learning with human feedback* (RLHF) is also emerging in interface adaptation – the system can occasionally ask the user “Was this input method convenient?” to get a richer reward signal. Over time, RL can personalize the interface to each user’s unique preferences and abilities, potentially yielding an **interface that learns from the user** just as the user learns to use the interface.

Balancing Exploration and Exploitation

A recurring theme in these approaches is the **exploration vs. exploitation** dilemma. In modality switching, exploitation means using the input method that the system currently believes is best for the context (to maximize immediate success), whereas exploration means trying a less-used or uncertain modality to potentially discover a better option for the future [10]. Simple heuristics don’t explicitly address this trade-off – they either *always* exploit the fixed policy or leave it entirely to the user to change things up. In contrast, bandit and RL methods naturally incorporate exploration. Algorithms like UCB (Upper Confidence Bound) choose actions that maximize a combination of expected reward *and* an exploration bonus for uncertainty [11] [12]. Practically, this might manifest as occasionally suggesting a modality that hasn’t been tried in the current context, or randomly routing a small fraction of inputs through an alternate modality even if one is working fine, just to verify that it remains optimal.

This balance is crucial because user interfaces and contexts are non-stationary. The “best” modality can change as the user becomes more skilled, as they grow tired, or as they move to a new environment. For example, a novice user might initially do better with simple click-based menus, but as they learn the system, speech or gesture could become faster. An adaptive system should **explore** those transitions at the right time. If it exploits too hard (never tries the new modality), it may stick the user with suboptimal interactions; if it explores too much (keeps toggling modes chaotically), it will frustrate the user. Successful systems find a balance – often gradually shifting more responsibility to the higher-bandwidth modalities once confidence in them improves. Empirical studies confirm that strategies addressing this trade-off yield better long-term outcomes. For instance, in Lin et al.’s contextual bandit gesture system, the algorithm’s exploratory nature meant it *kept looking* for better personalization even after initial improvements, resulting in continued gains between sessions [22] . Without that, adaptation might stall after a quick win.

In implementation, designers often set an **exploration rate** that decays over time – high exploration in early interactions (when the system knows little about the user/context) and more exploitation later once the system is well-tuned to the user. Yet, some ongoing exploration is usually maintained to detect changes. This is analogous to how an adaptive voice assistant might occasionally ask “Did you mean X?” or suggest “You can also say ‘...’” – it’s probing the user’s openness to new capabilities.

UX Design: Subtle and Non-Intrusive Modality Switching

No matter how intelligent the algorithm, the *user experience* can make or break multimodal interaction. A key principle is that modality switching or suggestions should be **subtle, context-appropriate, and preserve the user’s sense of control**. Some UX considerations and patterns include:

- **Minimize Mode Errors Silently:** If the system detects an error in one modality, it can sometimes correct or switch in the background without explicitly notifying the user. For example, if a voice command wasn’t understood, a smart AR assistant might seamlessly fall back to interpreting the user’s gaze as if they intended a selection, executing the likely action without making the user repeat themselves. This requires confidence – the system must be reasonably sure of the alternate interpretation. When done right, the user just perceives that the interface “understood them eventually” rather than showing a failure message. This non-intrusive recovery is better than interrupting with “Sorry, didn’t get that. Please click the button.” Many multimodal error-handling frameworks emphasize graceful degradation, where the system uses whatever partial input it got rather than failing completely.

- **Contextual Prompts and Ambient Cues:** If a switch must be suggested, it’s best done through gentle cues. Visual hints (an icon, highlight, or a tooltip that fades in) or slight haptic feedback can indicate an available modality. For instance, a smartwatch might give a gentle buzz when it’s an opportune moment to use a voice reply (e.g., when detecting that the user is driving and can’t easily tap). In AR HUDs, a small microphone icon might glow when ambient noise drops, subtly suggesting voice is now viable. Crucially, these cues should *not* nag the user. They should be easily ignored. User studies in XR have noted that conspicuous prompts can hurt immersion and comfort [23] . Therefore, designers strive for **ambient awareness**: making the system’s modality options visible in the interface (perhaps in a status bar or a quick-access menu) so that the user can notice alternatives at a glance, rather than the system explicitly interrupting to recommend a switch.

- **Personalization and Predictability:** Over time, the system can learn user preferences – some users might simply hate using voice and would rather not be prompted about it at all, while others may appreciate reminders. A user-facing settings panel or an onboarding questionnaire can allow people to express modality preferences or accessibility needs, which the system should honor in its policy (this can be integrated as a prior in a bandit's reward model, for example). Moreover, if the system does adapt automatically, it should communicate changes in a transparent way. One approach is the **"predictable magic"** guideline: the interface can change behavior, but in ways that are understandable in hindsight. For example, if the AR assistant stops accepting voice in a loud factory environment, it could show a small message like " Auto-disabled voice due to noise" in the corner. This gives the user context so they know it's not a bug and can resume voice when quiet again. Providing feedback about why a modality switch happened can increase user trust in adaptive systems.

- **Discrete & Low-Profile Modalities:** One way to make switching less intrusive is to have additional modalities that are *inherently subtle*. Researchers are actively exploring inputs that are less visible or audible to others. For instance, **microgestures** (tiny finger movements) that can be picked up by wearables or ring sensors offer a way to input commands without broad arm motions [4]. These could serve as a "stealth mode" input the system can suggest when big gestures would be socially uncomfortable. Likewise, **silent speech recognition** (lip-reading or subvocal speech via throat sensors) is being developed to allow voice-like input without actually speaking out loud [4]. In an adaptive system, if it knows you're in a meeting room with others, it might automatically switch to a silent speech mode (or at least give you the option) so you can still issue commands quietly. Commercially, we see early signs of this: e.g., Meta's prototype AR glasses include EMG wristbands for finger pinch detection – users can click by just intending a finger movement. Such modalities expand the palette for the system to choose from, making context-specific adaptation easier (since there's often a "quiet" alternative available).

Finally, it's important to evaluate these UX aspects in real deployments. Many research prototypes include user studies where participants experience the adaptive modality switching. Subjective feedback (through SUS, NASA-TLX, etc.) often highlights whether the switching was perceived as helpful or distracting [24] [25]. The consensus so far is that **timely, well-calibrated switching can reduce user workload and improve satisfaction**, but poorly executed switching (too frequent, unpredictable, or unjustified) can confuse users. Thus, the human factors side is just as critical as the algorithm.

Examples and Case Studies

To ground the discussion, here is a comparison of several representative approaches and systems:

Approach	Decision Mechanism	Context Used	Exploration Balance	Example Systems / References
Heuristic Fallback Rules	Fixed if-then rules for switching; predetermined priority order.	Specific cues (noise level, tracking loss, etc.) checked against thresholds.	No learning – always exploits the preset policy (no adaptation over time).	*Error-Aware Gaze*: auto-switch to head or controller if gaze fails [26]. Many AR UIs rely on user manual switching (no auto-exploration).
Confidence-Based Arbitration	Compute confidence/error for each modality each time; choose highest confidence modality or blend inputs weighted by confidence.	Real-time recognizer confidence, error rates, sensor quality metrics.	No explicit exploration (uses whichever modality appears most reliable at that moment).	*Weighted Pointer*: blends gaze & head input with weights based on predicted tracking error [26]. Early voice+pen systems used confidence voting.
Contextual Bandit (Online ML)	Treat modality choice as a multi-armed bandit problem; update choice policy with each interaction's reward.	Rich context vectors (environment, user state, recent outcomes, etc.) fed into bandit model (e.g. LinUCB).	Yes – algorithms (UCB, epsilon-greedy) ensure trying of less-certain modalities while exploiting the best-known option [10].	*Meta Adaptive Gestures*: LinUCB bandit personalized EMG gesture recognition, reducing errors over time [13] [12]. Widely used in BCI cursor control to pick optimal control signals [14].
Full RL Policy (Sequential)	Learn a policy (via Q-learning, DQN, etc.) mapping state to modality or UI adjustments to maximize long-term reward.	Comprehensive state (could include history, trends, cognitive load, etc.); not limited to instant context.	Yes – exploration during training (and sometimes ongoing). Often requires simulation or careful reward design to not perturb real users too much.	*Adaptive VR Interface (Carrow 2025)*: deep RL agent optimized combination of gesture/voice/gaze input, improving task time and satisfaction [17]. *NAMI (2025)*: used user cognitive state to adjust help modality, boosting performance [6] (hybrid approach with predefined logic + learning).

As seen above, the more advanced approaches (contextual bandits and RL) make use of broader context and **learn from experience**, offering personalization and adaptation that static methods lack. They incorporate exploration mechanisms to continue improving the interface as conditions change. On the other hand, they require more complex implementation – including gathering reward feedback and ensuring user comfort during adaptation – whereas heuristic methods are simpler and predictable.

Commercial systems today are only beginning to adopt these adaptive techniques. An example is **smart keyboard input suggestion** on mobile devices: modern smartphones can automatically suggest switching to voice dictation when detecting the user is driving (context: connected to car Bluetooth, motion sensors) – a heuristic but context-sensitive prompt. We anticipate future AR glasses and wearables will embed bandit-like personalization. In fact, Meta's research labs have demonstrated **prototype AR wristbands that learn gestures via bandit feedback**, aiming for calibration-free continuous learning [27] [13]. As these prototypes move toward products, consumers may experience UIs that quietly optimize themselves.

Conclusion

Multimodal interfaces in wearables and XR can greatly benefit from intelligent modality selection – improving robustness, efficiency, and user comfort by using "the right tool for the job" at any given moment. Contextual bandits and reinforcement learning provide a powerful framework for this adaptation, as evidenced by recent research that successfully applied these methods to gesture recognition, VR interaction, and other HCI domains [19] [13]. These learning approaches thrive on contextual data: by sensing the user's environment, physiological state, and interaction history, they tailor the interface in a way that static designs cannot, and they continue to refine their strategy through trial and feedback.

However, designing a successful adaptive multimodal system requires more than algorithms – it demands careful consideration of the *user experience*. The best systems strike a balance between automation and user agency, often by operating in a subtle, assistive manner. As Oviatt's early work on multimodal systems and more recent XR studies suggest, users appreciate when an interface quietly "has their back" (e.g., resolving ambiguities, recovering from errors) but can become frustrated if the system wrests control or behaves inconsistently. Therefore, **transparency, predictability, and the ability to override** are important features alongside adaptation.

Looking ahead, we see a convergence of these ideas in next-generation wearable and XR platforms. Imagine an AR headset that not only offers voice, gesture, eye, and brain inputs, but actively learns which you use most effectively in different contexts – essentially becoming a smart mediator between you and the digital world. Such a system could, for example, notice that when you're walking outside you prefer voice, but in a quiet office you switch to subtle hand gestures; it would then proactively present the voice UI when you step outdoors, and seamlessly shift to gesture mode in the office, without a manual toggle. Achieving this will rely on the continued maturation of on-device machine learning (for real-time bandit/RL decisions) and careful iterative design with user feedback.

In summary, adaptive modality selection is a promising frontier in HCI that combines the **strengths of multimodal interaction** (flexibility, naturalness) with **context-aware intelligence**. Early academic and prototype systems have demonstrated feasibility, showing improved interaction performance and user satisfaction through context-driven arbitration [19]. As these techniques make their way into consumer products, we expect more fluid and forgiving interfaces – ones that bend to the user's needs and context, rather than forcing the user to adapt to the interface. The ongoing challenge will be ensuring these smart interfaces remain trustworthy and inclusive, serving the general public and specialist users alike with minimal friction. With a multidisciplinary effort bridging HCI, AI, and design, the vision of an interface that *knows* when to listen, when to watch, and when to wait for your thought may soon become a mainstream reality.

[1] [2] [3] [4] [7] [20] [23] [24] [25] [26] \reviseTowards spatial computing: recent advances in multimodal natural interaction for XR headsets
https://arxiv.org/html/2502.07598v1

[5] [6] preprints.org
https://www.preprints.org/frontend/manuscript/5b4ac6406306b314fbdc11a4a1389d07/download_pub

[8] [9] Adaptive Multimodal User Interface Techniques for Mobile Augmented Reality: Frameworks, Modalities and User Interaction | Request PDF
https://www.researchgate.net/publication/394350811_Adaptive_Multimodal_User_Interface_Techniques_for_Mobile_Augmented_Reality_Frameworks_Modalities_and_User_Interaction

[10] [11] [12] [13] [14] [15] [16] [22] [27] A Contextual Bandits Approach for Personalization of Hand Gesture Recognition
https://arxiv.org/html/2509.08915

[17] [18] [19] [21] Paper Title (use style: paper title)
https://www.pspress.org/index.php/tcsm/article/download/254/204