AdaptaFace: Adversarial Assistance for Inclusive Facial Gesture Interaction

Siyu Zhang
s.zhang@bristol.ac.uk
University of Bristol
Bristol, United Kingdom

Yelu Gu
yelu.gu@bristol.ac.uk
University of Bristol
Bristol, United Kingdom

Xiaoyun Xu
xuxiaoyun@bza.edu.cn
Zhongguancun Academy
Beijing, China

Kirsten Cater
kirsten.cater@bristol.ac.uk
University of Bristol
Bristol, United Kingdom

Oussama Metatla
o.metatla@bristol.ac.uk
University of Bristol
Bristol, United Kingdom

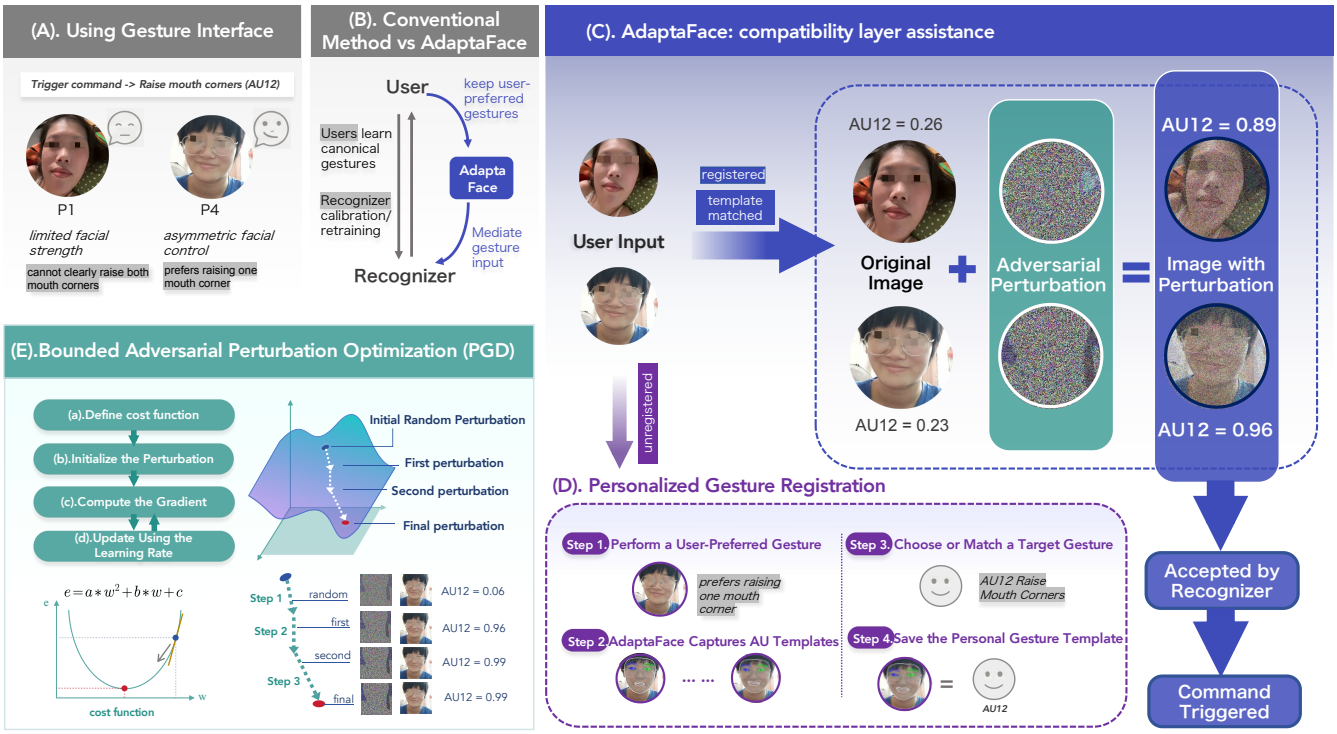

Figure 1: AdaptaFace overview: (A) user–recognizer mismatch, (B) input-side mediation, (C) runtime adversarial assistance, (D) personalized gesture registration, and (E) bounded perturbation optimization. AU12 denotes the lip corner puller.

Abstract

Facial gestures can enable access for people with motor impairments (PwM), yet recognizers may not reliably interpret individual facial movement patterns. Existing approaches either constrain agency by requiring predefined templates or impose repeated calibration, data collection, and computation through user-specific retraining. We present AdaptaFace, an input-level compatibility layer using bounded, proxy-dependent adversarial perturbations to translate user-compatible gestures into inputs for an unchanged facial action unit (AU)-based recognizer. Designed around six personalization principles informed by interviews with 12 PwM, AdaptaFace supports both recognition enhancement and gesture vocabulary extension. In a within-subject study with 16 participants, AdaptaFace increased task completion from 59.7% to 90.5% and reduced workload from 5.61 to 4.21. Offline validation showed strong steering with aligned perturbation generation and

UIST '26, November 2–5, 2026, Detroit, MI, USA

https://doi.org/10.1145/3830398.3830708

evaluation but weaker cross-recognizer transfer, defining the approach's boundary. These findings frame adversarial assistance as input-side accessibility mediation for personalized facial-gesture interaction.

CCS Concepts

• **Human-centered computing → Gestural input**; **Accessibility systems and tools**.

Keywords

Accessibility; Ability-based design; Ability-mediating design; Personalization; Facial gestures; Adversarial machine learning; Inclusive interaction

ACM Reference Format:

Siyu Zhang, Yelu Gu, Xiaoyun Xu, Kirsten Cater, and Oussama Metatla. 2026. AdaptaFace: Adversarial Assistance for Inclusive Facial Gesture Interaction. In *Proceedings of The 39th Annual ACM Symposium on User Interface Software and Technology (UIST '26).* ACM, New York, NY, USA, 18 pages. https://doi.org/10.1145/3830398.3830708

1 Introduction

Facial gestures such as head movements, blinks, and mouth actions provide an important access pathway for people with motor impairments (PwM) when hand-based input is difficult or impossible [3, 38, 42]. Because they can be captured using commodity cameras and require little or no upper-limb movement, they have long been seen as a promising modality for accessible interaction. Yet, facial-gesture interfaces commonly expose a small set of canonical movements. Users whose facial movements are subtle, asymmetric, personalized, or variable may not align reliably with those predefined gestures [41, 42]. When recognition depends on reproducing canonical expressions, many PwM must exaggerate, repeat, or alter their movements to make themselves legible to the recognizer. This shifts the burden of adaptation onto users, undermining both reliability and agency in everyday interaction [2, 5, 28, 38]. Interaction interfaces built on such recognizers can therefore produce allocative harms for PwM [41].

Prior personalization strategies commonly modify interaction mappings or adapt recognizers using user-specific data. Mapping approaches are bounded by the configurations an implementation exposes, while model adaptation requires calibration samples or demonstrations [10, 15, 29, 30, 37]. This motivates a third path that improves compatibility without changing model parameters or requiring users to conform to canonical gestures.

We investigate this third path through a compatibility layer that mediates between user expression and recognizer expectations. We operationalize this logic through adversarial assistance, repositioning adversarial methods not as threats to recognition systems, but as a mediation mechanism for accessible interaction. AdaptaFace uses a differentiable proxy to compute subtle perturbations to live facial input so that an unchanged AU-based recognizer is more likely to interpret gestures that match a user's own motor repertoire. The recognizer is not retrained, but the perturbation remains dependent on the proxy and target output space. This reframes personalization from requiring users to conform to recognizer-defined templates or retraining recognizers for each user toward connecting the two through a mediating layer.

We developed and evaluated AdaptaFace in three stages. We interviewed 12 PwM to derive six design principles for low-burden personalization. We then built a browser-based system with gesture registration, assistance tuning, and four personalization modes: Fixed Enhancement, Matched Enhancement, Fixed Extension, and Matched Extension. Finally, we evaluated it with 16 participants with motor impairments across four tasks and complemented the study with offline transfer analyses across recognizers and datasets.

We contribute: (1) an accessible interaction logic based on an input-level compatibility layer, which mediates between user expression and recognizer expectations through adversarial assistance rather than user conformity or model retraining; (2) AdaptaFace, a browser-based facial interaction system with user-facing interfaces for gesture registration, assistance tuning, and four personalization modes[1]; and (3) empirical evidence from formative interviews, a user study, and bounded offline validations showing improved performance, lower workload, greater perceived control, broader usable gesture vocabularies, and the current limits of cross-recognizer transfer.

2 Related Work

2.1 Facial Gesture Interaction as Accessible Input

Above-the-neck movements and actions, including head motion, gaze, blinks, eyelid actions, mouth movements, and tongue motion, have long been explored as accessible input for PwM, supporting communication, text entry, cursor control, and environmental interaction [2, 3, 8, 13, 17, 19, 27]. Because these inputs can often be performed without reliable hand use, they provide an important pathway for assistive interaction [32, 42]. Early systems frequently relied on specialized hardware such as eye trackers or instrumented tongue interfaces [17, 19, 27]. Commodity cameras and computer-vision toolkits now support real-time webcam and mobile facial analysis without specialist sensing hardware [1, 8, 12, 14, 16]. However, many existing recognizers expose a bounded vocabulary that users must reproduce reliably; recent elicitation studies instead show the value of gestures defined around users' own abilities [8, 32, 42]. This distinction matters for PwM: feasible movements can be low-amplitude or asymmetric, differ across individuals, and vary with ability and fatigue over time [39, 41]. Prior work also reports workload, discomfort, and frustration in above-the-neck control, while ability-based and design-justice scholarship cautions against making normative motor assumptions the basis of access [2, 5, 38, 41]. This leaves an unresolved interaction-layer gap: camera access alone does not make a canonical recognizer vocabulary compatible with a person's feasible movement repertoire.

[1] Implementation: https://github.com/SiyuZhang-zsy/AdaptaFace

2.2 Personalization in Recognition-Based Accessibility

Personalization has become a central strategy for addressing the mismatch between users' abilities and one-size-fits-all interaction systems. In assistive HCI, personalization is important because inclusive systems must adapt to diverse motor repertoires rather than require users to conform to predefined interaction norms [5, 35, 38]. Within facial-gesture and broader recognition-based accessibility, personalization spans at least two established strategies. One changes the interaction mapping or interface around a user's abilities without retraining the recognizer. Supple automatically generates interfaces around devices, tasks, preferences, and abilities; MotionBlocks remaps comfortable movements to required virtual-reality input; and gesture-elicitation studies derive vocabularies from movements that users define themselves [10, 32, 37, 42]. These systems show that personalization can occur through interface generation, remapping, or user-defined vocabularies, with each implementation defining which mappings, parameters, or recognition capabilities are available. A second strategy adapts a recognizer from user-specific data through calibration, fine-tuning, or user demonstrations [15, 29, 30]. Even few-shot methods rely on user-provided calibration samples or demonstrations [15, 29, 30]. When those samples include gaze data, their collection and storage can also raise biometric privacy concerns [6, 31]. For users whose feasible movements are ability-specific or fatigue-sensitive, providing personal movement data is therefore a consequential design consideration [32, 39].

Together, these approaches alter either the interaction mapping or the model. A less explored alternative is to improve compatibility at the recognizer input, mediating between user expression and recognizer expectations without changing model parameters.

2.3 Adversarial Methods as Input-Side Adaptation

Adversarial perturbations have been widely studied in machine learning as small input modifications that can shift a model's prediction by exploiting its decision boundaries [11, 33]. Most prior work has framed adversarial examples as threats to robustness and security, motivating extensive research on attacks, defenses, and model stability [4, 26]. Perturbation-based methods have also been used for interpretability, while adversarial reprogramming has repurposed fixed models for new tasks [7, 9]. These efforts show that controlled input transformations can be used not only to expose vulnerabilities but also to probe or repurpose model behavior.

The accessibility work reviewed above personalizes interaction mappings or models; it does not examine adversarial input transformation as a personalization mechanism. Model-adaptation approaches instead personalize recognition from user-specific calibration samples or demonstrations, even when few-shot methods reduce the amount required [15, 29, 30]. Adversarial reprogramming provides a technical precedent for another possibility: adjusting the input while keeping the target model's parameters fixed [7]. AdaptaFace applies this input-side logic to accessibility, seeking to make an existing recognizer more compatible with a user's expression patterns. This perspective complements ability-based design, which asks systems to adapt to users' abilities rather than the reverse [38]. It also connects to ability-mediating design, which distinguishes adapting a system to an ability from mediating an existing sensorimotor ability so that it can enable a new skill or interaction [34]. At the recognizer input, the system preserves the user's performed movement while mediating the image presented to the recognizer. We use this connection as a conceptual lens, not as evidence that the present implementation covers the broader sensorimotor-reality design space.

Table 1: Demographic and functional characteristics of formative-study participants with motor impairments (FP1–FP12).

ID	Age	Gender	Health Condition	Since	Functional Traits
FP1	32	F	SMA II	6 mo.	Ir, Pr, Fa, Dig, Dif
FP2	31	M	SCI (C1)	10 yrs	Ir, Fa, Dig, Dif, Tre
FP3	36	F	SCI (C5)	5 yrs	Ir, Fa, Dig, Dif
FP4	18	F	Stroke (hemi.)	5 yrs	Ir, Fa, Dig, Dif
FP5	32	M	Brainstem Tumor (hemi.)	3 yrs	Fa, Dif
FP6	26	M	Cerebral Palsy	Birth	Ir, Fa, Dig, Dif
FP7	36	F	SCI (T1–T4)	5 yrs	Ir, Fa, Dig
FP8	31	M	Poliomyelitis (para.)	28 yrs	Pr, Fa, Dig
FP9	33	M	Cerebral Palsy	Birth	Ir, Pr, Dig, Dif, Tre
FP10	28	M	Cerebral Palsy	Birth	Ir, Fa, Dig, Dif, Tre
FP11	24	F	Transverse Myelitis	Birth	Ir, Fa, Dif
FP12	32	F	CMT2S	Birth	Ir, Fa, Dig, Dif

Abbreviations: SMA = Spinal Muscular Atrophy; SCI = Spinal Cord Injury; CMT = Charcot–Marie–Tooth disease; hemi. = Hemiplegia; para. = Paraplegia.
Functional traits: Ir = Irreversible; Pr = Progressive; Fa = Rapid fatigue; Dig = Difficulty gross motor; Dif = Difficulty fine motor; Tre = Tremor.

3 Formative Study and Design Principles

3.1 Participants and Method

To ground the design of AdaptaFace, we conducted a formative interview study with 12 PwM (6 women, 6 men; aged 18–36) (Table 1). Participants represented diverse motor conditions and reported varying levels of facial control, asymmetry, fatigue, and movement consistency. We refer to them as FP1–FP12 to distinguish them from evaluation-study participants. We recruited them through social media platforms, non-profit disability organizations, and personal networks. All participants provided informed consent and received a 15 USD honorarium.

Interviews explored three topics: prior experiences with facial and above-neck interaction, perceived barriers in recognizer-defined gesture systems, and expectations for personalization. Remote sessions lasted 45–60 minutes and were recorded with consent. Participants first discussed their abilities and prior interaction experiences, then tried atomic facial actions and reflected on their comfort, feasibility, and distinctness, and finally proposed gestures for example commands using those actions only as a loose reference. We analyzed the interviews using thematic analysis to identify design-relevant patterns related to gesture performance, personalization burden, and desired system behavior. Our aim was not to claim that all reported accessibility barriers were new; rather, the study translated barriers documented in prior work and participant-specific experiences into design principles and system requirements. Further procedural details are provided in Appendix B.2.

3.2 Findings

F1. Canonical gestures often failed to fit participants' lived motor repertoires. Participants described difficulty reproducing standardized facial gestures due to diverse motor constraints; for instance, FP5 noted, "*At my worst, half my face was numb; it's hard to make expressions*", while FP1 explained, "*My muscles are weak; it's hard for me to lift the corners of my mouth*". As a result, their comfortable gestures were asymmetric, subtle, and did not necessarily resemble the canonical expressions recognizers were designed for, leading to frequent mismatches in recognition.

F2. Repeated calibration and retraining imposed disproportionate burden. Participants did not reject personalization itself, but many described repeated demonstration, recalibration, and adaptation as tiring and frustrating; as FP4 noted, "*It doesn't really pick up my expressions unless I exaggerate them, which is tiring*". This burden was further exacerbated when performance varied across posture, fatigue, or day-to-day condition. For those whose movements differed most from canonical templates, the burden of "making the recognizer work" was often highest.

F3. Participants wanted personalization that preserved their own expressive habits and sense of control. Beyond accuracy alone, participants emphasized the importance of using gestures that felt natural, feasible, and socially acceptable to them; as FP11 described, "*If I can nod to confirm, that's respectful—it's the same as how I would signal to another person. It feels natural, not like I'm performing something strange for the computer*". At the same time, participants expressed a need to better understand how their actions were interpreted by the system, suggesting a desire for greater visibility and control rather than opaque automatic adaptation. Together, these accounts suggest that inclusive personalization should improve reliability without removing user agency.

3.3 Design Principles

Based on these findings, we derived six design principles for inclusive facial-gesture personalization. Together, these principles point toward a different locus of personalization: a mediating mechanism at the input level that preserves users' own expressive habits while helping existing recognizers interpret them more reliably. Specifically, the system should: (1) support user-compatible gestures rather than only canonical templates; (2) minimize user burden during personalization; (3) enable user-defined gesture vocabularies; (4) preserve user control over personalization; (5) preserve compatibility with existing recognition pipelines; and (6) support low-effort and socially acceptable interaction.

Detailed descriptions of each design principle and how they informed the system design are provided in Appendix B.3.

4 AdaptaFace: Compatibility Layer

To instantiate the design principles derived from our formative study, we built AdaptaFace, a real-time facial interaction system organized around a compatibility layer between user expression and recognizer expectations. Rather than changing the downstream recognizer itself, AdaptaFace changes the input pathway into it. This allows users to perform gestures that are feasible and comfortable for them while preserving compatibility with existing recognizers and applications. We describe the compatibility-layer logic, the system pipeline, the adversarial-assistance mechanism, the user controls exposed by the interface, and the four personalization modes supported by the system.

4.1 Compatibility-Layer Logic

As explained in the related work, AdaptaFace explores a third strategy of adaptation: *input-side mediation*. A compatibility layer sits between live facial input and the downstream recognizer, transforming the incoming frame so that the recognizer is more likely to interpret a user-compatible gesture as the intended command (Panel B in Figure 1). The recognizer, its label space, and the application-level command mapping remain unchanged.

We instantiate this compatibility layer through *adversarial assistance*: a bounded, targeted perturbation applied to live facial input to steer recognition toward a registered target output. The goal is not to attack the recognizer in an uncontrolled way, but to mediate between a user's expressive repertoire and the recognizer's predefined decision space while keeping the perturbation visually subtle. This mediation supports two functions. First, it enables *recognition enhancement*, improving reliability for gestures that are already represented in the recognizer's native output space. Second, it enables *gesture vocabulary extension*, allowing a user-defined gesture outside that native vocabulary to be mediated toward an existing recognizer output without retraining the model.

4.2 System Overview

AdaptaFace has two phases: registration and runtime interaction. During registration, the user either selects a gesture from a predefined set or records custom gestures and associates them with an intended command. Depending on the personalization mode, the target recognizer output is either predefined or determined through a compatibility-based matching procedure during registration.

Each registered gesture is therefore associated with a target recognizer output before runtime interaction begins. In the fixed modes, this target is specified in advance. In the matched modes, AdaptaFace compares the registered gesture against a set of candidate recognizer outputs during registration and selects the most compatible target once. Runtime interaction does not dynamically switch among multiple targets.

The unit of personalization is therefore a binding among a feasible gesture, a stable internal recognizer target, and an existing application command, rather than a retrained recognizer.

During runtime, AdaptaFace captures live facial frames and checks whether the incoming motion matches one of the registered gesture patterns. When a registered gesture is detected, the compatibility layer computes a bounded perturbation and applies it to the current frame before passing the assisted frame to the downstream recognizer (Panel C in Figure 1). The assistance is applied toward the target previously assigned to the detected gesture, and the recognizer's output is then mapped to the corresponding application command as in a standard recognition pipeline. AdaptaFace thus personalizes interaction without modifying the recognizer itself or requiring model retraining.

4.3 Matching, Mediation, and Direct Triggering

Gesture matching and image mediation serve different roles in this pipeline. Matching determines which registered pattern is active and retrieves its internal target. That target is control information within AdaptaFace; it is not sent to the downstream recognizer as a predicted gesture label. Adversarial assistance then uses the target to modify the recognizer-facing image, and only the resulting image is passed downstream.

This distinction provides a recognizer-facing matching-only comparison. If the perturbation is disabled while the matched target remains internal, the downstream recognizer receives the same raw frame as in Baseline; matching alone therefore cannot alter that recognizer's AU output. A different architecture could use the matched label to trigger an application command directly. Such direct triggering is a valid and potentially simpler option when an application exposes a command application programming interface (API) and can adopt AdaptaFace-specific events. It is not the same integration setting evaluated here: it bypasses the image recognizer and changes the application input path. AdaptaFace instead addresses settings in which the downstream image-based recognizer and command mapping remain unchanged.

At runtime, let $\mathcal{R} = \{r_1, \ldots, r_M\}$ denote the registered gesture trajectories, each associated with a target recognizer output assigned during registration. Each incoming frame is represented by an unnormalized eight-dimensional AU activation vector $\mathbf{a}_t$, and the matcher maintains the causal window $A_t = \{\mathbf{a}_{t-W+1}, \ldots, \mathbf{a}_t\}$. For each trajectory $r_j = \{\mathbf{r}_{j,1}, \ldots, \mathbf{r}_{j,W}\}$, it computes the mean aligned Euclidean distance $D_j(A_t, r_j) = W^{-1} \sum_{i=1}^{W} \|\mathbf{a}_{t-W+i} - \mathbf{r}_{j,i}\|_2$ and similarity $s_j(A_t, r_j) = 1/(1 + D_j)$. The highest-similarity template is selected, with ties resolved deterministically by template identifier, and activates only when $s_j \geq \tau_{\text{match}}$. The matcher exposes T as a configurable stability requirement; the study configuration used $W = 1$, $\tau_{\text{match}} = 2/3$ (equivalently, Euclidean distance at most 0.5), and $T = 3$, so the same template had to remain top-ranked and above threshold for three consecutive frame-level windows. Once confirmed, the detected trajectory retrieves its previously assigned internal target and conditions frame-level perturbation generation; the matching label itself is not passed to the downstream recognizer. Appendix A.3 provides the expanded onset-detection description.

4.4 Adversarial Assistance Mechanism

To generate assistance efficiently, AdaptaFace uses a differentiable proxy model to estimate how small pixel-space changes affect recognizer outputs. We instantiate this process using a projected gradient descent (PGD)-style targeted optimization procedure with a random start. Given an input frame $x \in \mathbb{R}^{H \times W \times C}$ and a target recognizer output y^* associated with the detected registered gesture, the system iteratively computes a bounded perturbation δ that reduces the proxy-model loss for y^* while constraining $\|\delta\|_\infty \leq \epsilon$. We use a small number of iterations in practice to preserve real-time performance and keep the perturbation visually subtle. Formally, the perturbation is computed as

$$\delta^* = \arg \min_{\|\delta\|_\infty \leq \epsilon} \mathcal{L}\big(f_{\text{proxy}}(x + \delta), y^*\big), \tag{1}$$

where $f_{\text{proxy}}(\cdot)$ is a differentiable proxy model, $\mathcal{L}$ is a targeted loss, and ϵ bounds perturbation magnitude.

The proxy model is used to generate perturbations; the downstream recognizer's parameters and output vocabulary remain unchanged. During registration, AdaptaFace determines which recognizer output should serve as the target for each registered gesture; during runtime, it applies frame-level assistance toward that assigned target whenever the corresponding gesture is detected. The perturbation is nevertheless proxy- and output-space-dependent. We instantiate the mechanism with differentiable facial-AU models and evaluate cross-recognizer transfer separately in Section 6. Additional implementation details are provided in Appendix A.1.

4.5 Four Personalization Modes

AdaptaFace defines four personalization modes from two axes: whether interaction stays within or extends beyond the recognizer vocabulary (*enhancement* vs. *extension*) and whether the assistance target is predefined or selected through compatibility-based matching during registration (*fixed* vs. *matched*). In fixed modes, AdaptaFace steers recognition toward a predefined target recognizer output. In matched modes, AdaptaFace compares the user's registered gesture against a set of candidate recognizer outputs during registration and selects the most compatible target once. During runtime, assistance is applied to the target assigned during registration. In both cases, the downstream recognizer is unchanged, and the compatibility layer adapts the input pathway that leads into it. Together, the four modes span fixed versus matched target assignment and enhancement versus extension.

Fixed Enhancement. Users interact through a predefined gesture set specified by the interface, with fixed gesture–command mappings. AdaptaFace applies assistance toward a fixed recognizer-supported target, improving reliability for gestures that already exist within the interface vocabulary. This mode is closest to a conventional recognizer-centric pipeline, in which neither the gesture vocabulary nor the mapping is personalized.

Matched Enhancement. This mode is designed for gesture-selectable interfaces. During registration, AdaptaFace compares the user's performed gesture against the set of gestures supported by the interface and selects the candidate gesture that is most compatible. The selected match is then used as the target recognizer output during runtime interaction. This mode allows personalization within a predefined vocabulary by identifying the interface-supported gesture that best aligns with the user's performance, while keeping the available gesture set bounded by the designer.

Fixed Extension. Users define and record their own gesture–command bindings. Each custom gesture is associated with a fixed target recognizer output selected during registration. This mode extends interaction beyond a closed predefined vocabulary while keeping the assistance target stable during runtime. It supports user-defined input without requiring changes to the downstream recognizer.

Matched Extension. Users define and record their own gestures, but instead of manually binding each gesture to a fixed target, the

system compares the registered gesture with candidate recognizer outputs during registration and selects the most compatible target. This mode combines user-defined input with system-selected target matching and runtime input-side mediation toward that selected target. It is intended to support subtle, asymmetric, or only partially aligned gestures and enables gesture vocabulary extension.

4.6 Parameter Settings

The compatibility layer is governed by a small set of optimization parameters. In the study, these were exposed as lightweight user-facing controls; in the offline validations, they were fixed for consistency ($\epsilon = 8/255$, $\alpha = 3/255$, $K = 3$). These controls allowed participants to balance steering effectiveness, responsiveness, stability, and visual subtlety without requiring developer intervention.

Table 2 shows the user-facing controls and intended interaction effects. Higher strength, precision, and iteration settings increase the likelihood of successfully steering recognition toward the intended target, but may also introduce stronger visual effects, less smooth updates, or higher latency. The recognition-speed preset and gesture cooldown let users trade off responsiveness against stability and accidental repeated triggers. Internally, assistance strength, assistance precision, and iteration steps correspond to the perturbation bound ϵ, update step size α, and number of optimization steps K, respectively, but are exposed to participants through mapped interface ranges rather than raw optimization values. Figure 2 shows how these controls are surfaced in the interface.

4.7 Implementation and Deployment

We implemented AdaptaFace as a modular framework with a Python-based core that separates live input capture, compatibility-layer processing, and downstream recognition. In the current implementation, the compatibility layer is instantiated with differentiable proxy models based on OpenFace 3.0 and OpenGraphAU [25]. In the online study configuration, the same OpenFace-based AU model supplied both the perturbation-generation objective and the post-assistance AU scores; the online evaluation therefore tested an aligned proxy/evaluator setting rather than an independent downstream recognizer. Participants accessed a browser-based client that supported gesture registration, assistance tuning, and real-time interaction. The study interface was exposed remotely via ngrok, while all computation ran on a local laptop equipped with an NVIDIA RTX 3060 graphics processing unit (GPU). Under this setup, the system maintained end-to-end latency below 80 ms per frame, supporting real-time assistance in the study setting. The same hardware configuration was also used for local testing. This deployment demonstrates an insertion point immediately before an AU-based recognizer; support for other recognizer families depends on a compatible proxy and target space and is not established by the present implementation.

5 Evaluation with participants with motor impairments

We evaluated AdaptaFace with participants with motor impairments to assess how well input-level adversarial assistance supports facial-gesture interaction across the four personalization modes. Our goals were: (G1) to quantify the performance benefits of input-level assistance across tasks and gesture types, (G2) to evaluate usability, cognitive effort, and perceived control during interaction, and (G3) to understand user preferences, perceived mode suitability, and how different modes aligned with individual abilities.

5.1 Study Design

We used a within-subjects design with three independent variables: *Technology* (Baseline vs. AdaptaFace), *Task* (Swipe, Click, Zoom In, Zoom Out), and *Personalization Mode* (Fixed Enhancement, Matched Enhancement, Fixed Extension, and Matched Extension). This yielded a $2 \times 4 \times 4$ factorial design. Each Technology $\times$ Task $\times$ Mode task block contained five action repetitions, resulting in 32 planned task blocks and 160 planned action repetitions per participant, or 512 task blocks and 2,560 action repetitions overall. We counterbalanced the sequence of technology conditions across participants, balanced task order using a Latin square, and randomized mode order within each Technology $\times$ Task block. The design compares the complete Baseline and AdaptaFace pipelines across personalization modes; it does not independently ablate matching, target selection, and perturbation generation.

5.2 Participants

We recruited 16 adults with motor impairments that limited reliable hand use but preserved some above-neck control (N=16; 11 female, 5 male; age 19–40, mean M=28.9, standard deviation SD=5.1) (Table 3). Participants were recruited via disability organizations, online support communities, and word of mouth. Inclusion required (i) reduced or unreliable upper-limb function for everyday device use and (ii) ability to perform at least one facial or head gesture. Participants represented diverse impairment etiologies, including cerebral palsy, spinal cord injury, stroke, spinal muscular atrophy, Duchenne muscular dystrophy, Charcot–Marie–Tooth disease, spina bifida, Behçet's disease, and other paralysis. 14 participants used wheelchairs, and all reported prior experience with above-neck input such as gaze, head movement, or facial actions. 5 participants had also taken part in the formative study. All sessions were conducted remotely to remove travel barriers. The study allowed breaks, and supported alternatives such as typing. Both the formative study and the subsequent evaluation were approved by the ethics committee of the University of Bristol. All participants provided informed consent, and each evaluation participant received a 15 USD honorarium.

5.3 Apparatus, Tasks, and Procedure

Participants completed the remote study using their everyday phones or laptops and built-in webcams, connecting to AdaptaFace through a secure browser client from their own environments and preferred postures. The system was a zero-install web interface using webcam input to provide real-time input-side assistance and

Table 2: User-facing controls in the participant study. Values refer to interface-level controls rather than raw optimization parameters. In the offline validations, perturbation settings were fixed to $\epsilon = 8/255$, $\alpha = 3/255$, and $K = 3$.

Control	Role	Range / Default	Effect
Assistance strength (ϵ)	Perturbation budget	1–20 / 5	steering ↑; too high → overshoot
Assistance precision (α)	Update step size	1–10 / 5	aggressiveness ↑; too high → less smooth
Iteration steps (K)	Refinement steps	1–10 / 3	accuracy ↑, latency ↑
Recognition speed preset	Response–stability balance	1–5 / 3	faster response ↔ calmer behavior
Gesture cooldown (ms)	Trigger interval	100–2000 / 400	repeats ↓; rapid actions slower

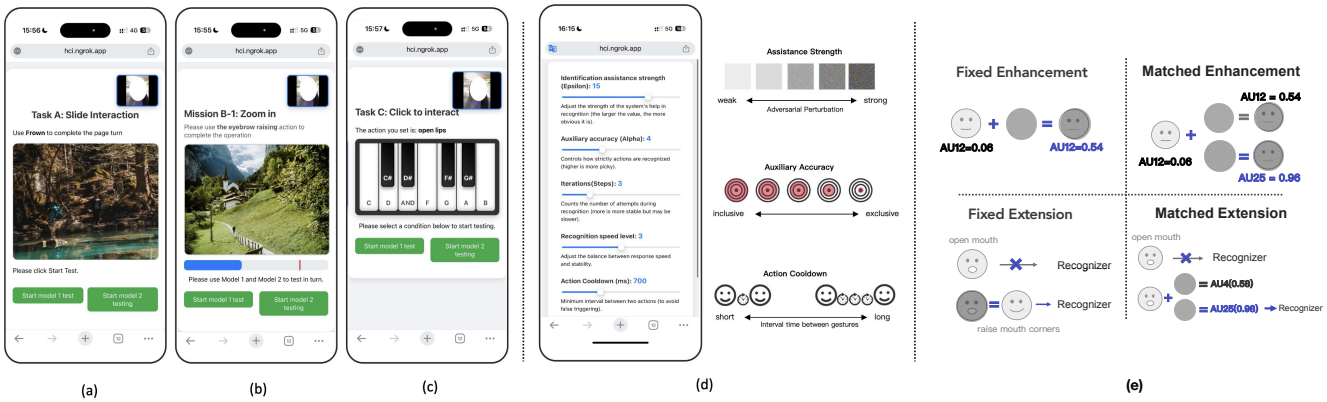

Figure 2: Evaluation interfaces and modes: (a) Swipe, (b) Zoom, (c) Click, (d) assistance controls, and (e) Fixed Enhancement, Matched Enhancement, Fixed Extension, and Matched Extension.

Table 3: Participant characteristics for the evaluation study ($N = 16$). FS identifies participants who also took part in the formative study.

ID	Age	Gender	Condition	Device	Wheel.	FS
EP1	32	Female	SMA2	Phone	Yes	FP1
EP2	36	Female	SCI(C5)	Phone	Yes	FP3
EP3	37	Male	DMD	Phone	Yes	–
EP4	19	Female	Stroke	Phone	No	FP4
EP5	28	Female	SMA2	Phone	Yes	–
EP6	32	Female	CMT2S	Phone	Yes	FP12
EP7	24	Female	SMA2	Computer	Yes	–
EP8	40	Female	Paralysis	Phone	Yes	–
EP9	27	Female	SMA3	Phone	Yes	–
EP10	28	Female	SMA	Computer	Yes	–
EP11	25	Female	SCI	Phone	Yes	–
EP12	26	Male	CP	Phone	No	FP6
EP13	27	Male	SCI(T3)	Phone	Yes	–
EP14	28	Male	SMA3	Phone	Yes	–
EP15	35	Female	SB	Phone	Yes	–
EP16	30	Male	BD	Phone	Yes	–

Abbreviations: SMA = Spinal Muscular Atrophy; SCI = Spinal Cord Injury; DMD = Duchenne Muscular Dystrophy; CMT = Charcot–Marie–Tooth disease; CP = Cerebral Palsy; SB = Spina Bifida; BD = Behçet's Disease; Wheel. = Wheelchair; FS = Formative Study.

adapted feedback. We logged lightweight metadata required for analysis, including timestamps, task events, recognition outcomes, confidence scores, and gesture–command mappings.

We evaluated AdaptaFace on four interaction tasks: *Swipe* (continuous navigation), *Click* (discrete selection), *Zoom In*, and *Zoom Out* (bidirectional scaling) (Figure 2). These tasks capture different interaction demands for repetition, precision, and continuous control.

Each session lasted 30–60 minutes depending on pace and breaks. After consent and setup, participants joined a video call, received an introduction to the task interfaces, and completed a practice round. Participants then completed all experimental blocks. In modes other than *Fixed Enhancement*, they could select or define gestures that suited their abilities; in matched modes, target recognizer outputs were assigned automatically during registration. At the start of each block, participants could review or re-record gestures if needed. After each block, they rated ease of use and perceived control. After the session, they completed the System Usability Scale (SUS) and the NASA Task Load Index with raw scoring (NASA Raw TLX) separately for Baseline and AdaptaFace, followed by a short semi-structured exit interview on gesture preferences, perceived control, workload, and social acceptability. To reduce fatigue, participants were encouraged to take breaks between tasks and could pause at any time or split the session across two sittings if needed.

5.4 Measures

We collected objective performance metrics, subjective experience measures, and records of gesture creation and mappings in personalization modes.

Table 4: Performance by technology. Δ shows AdaptaFace minus Baseline (units follow each column); Baseline is the unmodified OpenFace-3.0 recognizer with default AU thresholds.

	CR % (SD)	TCT s (SD)	SNT s (SD)	Conf. (SD)
Baseline	59.7 (44.7)	16.6 (16.7)	37.7 (151.3)	0.65 (0.43)
AdaptaFace	90.5 (25.6)	13.4 (9.0)	15.9 (16.1)	0.89 (0.28)
Δ (A−B)	+30.8	-3.2	-21.8	+0.24

Objective performance. For each task block, we logged task completion time (TCT), completion rate (CR), success-normalized task time (SNT), and event-level recognition confidence. TCT measured the elapsed time required to complete the assigned interaction sequence. CR was the proportion of required action repetitions successfully completed within the block. For inferential analysis, we additionally derived full-block completion, a binary indicator of whether all required repetitions in a block were completed. SNT normalized task time by the number of successful repetitions, enabling comparison when completion was partial. Recognition confidence was computed at the event level rather than per frame.

Subjective experience. After each block, participants provided two 5-point Likert ratings on ease of use and perceived control. After completing all tasks, they filled out the System Usability Scale (SUS) and the NASA Raw TLX separately for Baseline and AdaptaFace.

Gesture creation and mappings. In personalization modes, we recorded the chosen gesture type, the bound command, whether the gesture extended beyond the recognizer's exposed AU vocabulary, and the associated parameter settings. This enabled us to analyze gesture diversity, vocabulary extension, and how participants tuned assistance.

Qualitative feedback. A semi-structured exit interview gathered reflections on gesture preferences, perceived strengths and weaknesses of AdaptaFace, trust and feedback, social acceptability, and long-term use.

5.5 Data Analysis

We analyzed binary full-block completion using a binomial generalized estimating equation (GEE) with a logit link, and continuous repeated-measures outcomes using linear mixed-effects models (LMMs) with participant-level random intercepts. Task completion time (TCT), success-normalized task time (SNT), ease, and perceived control were analyzed within this repeated-measures framework, while CR percentages were summarized descriptively. SUS and NASA Raw TLX, which were collected per technology, were compared descriptively at the participant level between Baseline and AdaptaFace. No Technology × Task or Technology × Mode interaction reached significance; we therefore focus on Technology main effects, while task- and mode-level breakdowns are interpreted descriptively. Model equations and fixed-effect tables are reported in Appendix Section B.1.

Qualitative interview responses were analyzed using thematic analysis. Two researchers reviewed transcripts and notes iteratively, grouped recurring observations into themes, and used these themes to interpret participants' gesture preferences, perceived control, workload, and social acceptability. In the main paper, we report qualitative findings primarily to contextualize the quantitative results rather than as a standalone qualitative contribution.

5.6 Results

Results follow G1–G3. We first report objective performance and mode-level patterns, then subjective experience, and finally gesture-vocabulary extension. Qualitative accounts then examine how participants experienced the personalization choices underlying those outcomes.

5.6.1 Overall Objective Performance. Across tasks and personalization modes, AdaptaFace improved all objective measures (Table 4). Mean within-block CR increased from **59.7%** to **90.5%**; a GEE on binary full-block completion showed a significant Technology effect (β=1.12, Wald z=11.86, p<.001). TCT decreased from **16.6 s** to **13.4 s** (LMM: β=−3.18, z=−7.92, p<.001, 95% confidence interval (CI) [−3.97, −2.40]); SNT decreased from **37.7 s** to **15.9 s** (β= − 12.16, z= − 12.63, p<.001, 95% CI [−14.05, −10.27]). Mean event-level confidence increased from **0.65** to **0.89**.

Completion improved in every mode (Table 5): *Fixed Enhancement*, **77.3%–94.8%**; *Matched Enhancement*, **77.6%–92.7%**; *Fixed Extension*, **27.9%–82.4%**; and *Matched Extension*, **55.8%–91.9%**. The enhancement modes improved reliability while retaining the recognizer's existing vocabulary, whereas the largest descriptive changes appeared in the extension modes, where personalized gestures were unlikely to succeed without assistance. SNT showed the same pattern. Because no Technology × Task or Technology × Mode interaction was significant, these breakdowns are descriptive rather than confirmed effect-size differences.

Completion and confidence improved across all four tasks, whereas time measures did not improve uniformly (Appendix Table 13). *Zoom In* was the most variable Baseline task, while *Click* had the highest completion and lowest AdaptaFace task time. Together, the mode and task patterns show two complementary outcomes: more reliable use of recognizer-supported gestures and practical use of personalized gestures outside the exposed vocabulary.

Runtime template matching, used in all four AdaptaFace modes and distinct from registration-time target matching in the Matched modes, was activated at least once in **95.6%** of the analyzed AdaptaFace task blocks. Among successful runtime matching events, Euclidean matching distance had a median of **0.1067** (interquartile range (IQR) [0.0256, 0.2683]), a mean of **0.1549** (SD = 0.1469), and a range of [**0.0000**, **0.4998**].

5.6.2 Subjective Experience. AdaptaFace increased ease from **3.70** to **4.35** (β=0.67, z=13.12, p<.001, 95% CI [0.57, 0.78]) and perceived control from **3.15** to **4.27** (β=1.12, z=20.48, p<.001, 95% CI [1.02, 1.23]). SUS increased from **47.3** to **70.8**, and NASA Raw TLX decreased from **5.61** to **4.21** (Table 6). Workload reductions were broad rather than isolated to one dimension: cognitive, physical, temporal, effort, and frustration scores decreased, while perceived performance improved. These converging measures indicate that

Table 5: Performance by mode (mean (SD)).

Mode	CR %		TCT s		SNT s		Conf.	
	Baseline	AdaptaFace	Baseline	AdaptaFace	Baseline	AdaptaFace	Baseline	AdaptaFace
Fixed Extension	27.9 (40.8)	82.4 (33.0)	22.0 (26.6)	16.6 (10.7)	135.4 (380.8)	23.5 (26.9)	0.38 (0.45)	0.82 (0.36)
Matched Extension	55.8 (45.4)	91.9 (24.1)	14.9 (10.2)	11.7 (8.7)	30.9 (48.3)	13.4 (10.4)	0.65 (0.43)	0.91 (0.24)
Fixed Enhancement	77.3 (35.2)	94.8 (17.2)	15.6 (12.3)	12.6 (7.8)	17.7 (17.9)	13.9 (9.1)	0.83 (0.31)	0.92 (0.21)
Matched Enhancement	77.6 (37.7)	92.7 (24.5)	13.9 (11.4)	12.5 (7.7)	16.2 (14.8)	13.3 (8.9)	0.76 (0.35)	0.89 (0.26)

Table 6: System-level scales. Higher is better for SUS; lower is better for NASA Raw TLX.

(a) SUS (0–100)

System	Mean	SD	Median
Baseline	47.3	15.7	50.0
AdaptaFace	70.8	14.7	72.5

(b) NASA Raw TLX

System	Mean	SD	Median
Baseline	5.61	1.54	5.83
AdaptaFace	4.21	1.56	4.50

(c) NASA Raw TLX subscale deltas (AdaptaFace − Baseline)

Subscale	Cognitive	Physical	Temporal	Effort	Frustration	Performance
Δ	−0.87	−1.33	−1.27	−1.00	−2.00	+1.93

participants experienced the assisted pipeline as both easier to operate and more controllable. Appendix B reports task- and mode-level ratings.

5.6.3 Gesture Vocabulary Extension. All **16 participants** introduced personalized gestures in extension modes. The resulting repertoire included oral/lip actions such as *Pout*, *Lip Press*, and *Mouth Widen*; ocular/gaze actions such as *Close Right Eye*, *Eye Look Up*, and *Half-squinted Eyes*; and cheek- and tongue-based gestures. Oral/lip gestures were most frequent (**40** instances), followed by ocular/gaze/brow (**38**), cheek (**5**), and tongue (**4**).

Several gestures, including *Pout*, *Lip Press*, and *Close Right Eye*, were reused across tasks. Participants were therefore developing small personal repertoires rather than selecting isolated substitutions for individual tasks. This distinguishes extension from enhancement: the enhancement modes improved reliability within the recognizer's vocabulary, while the extension modes broadened which movements could function as input at all. Appendix B reports the full inventory.

5.7 Qualitative Feedback

Interviews showed that AdaptaFace changed not only how reliably gestures were recognized, but also which movements participants felt they could comfortably and confidently use as input. We organize these accounts into three themes connecting performance to the experience and tradeoffs of personalization.

5.7.1 AdaptaFace preserved participants' own gesture abilities and preferences. Participants valued gestures aligned with their motor abilities rather than canonical expressions. Common preset gestures could be physically demanding or unnatural; one participant noted that *"for many people with severe motor impairments, opening the mouth wide is very tiring."* EP1 explained the benefit of extension more directly: *"Even if I can't open my mouth wide, I can still use a small version in custom mode,"* while EP6 noted, *"Everyone has different strengths; customization lets you pick what you can actually do."* Rather than treating deviation from a canonical form as failure, personalization preserved movements that were sustainable and personally appropriate.

5.7.2 AdaptaFace reduced adaptation burden and improved perceived ease of use. Participants contrasted AdaptaFace with Baseline in terms of both physical and cognitive effort. Baseline often required exaggerated expressions (EP3: *"You have to open your mouth really wide"*), whereas EP4 described AdaptaFace as *"less tiring."* EP13 highlighted a different burden: *"If you set it up for me, I might not remember...My own gestures are much easier to remember."* Self-defined mappings therefore reduced the need to repeatedly exaggerate movements and the need to remember system-imposed gestures. This helps contextualize the improvements in ease, control, and workload as consequences of a better fit between interaction demands and participants' existing habits.

Perceived control did not arise from automation alone. Mirror views, live labels, and immediate responses helped participants judge whether a gesture had been recognized, understand the current gesture–command mapping, and decide whether to re-record a gesture or adjust assistance. This feedback made personalization legible rather than invisible. At the same time, some controls and parameter names felt technical, creating a tension between configurability and setup burden. Adjustable assistance supported agency when participants could see its effects, but additional control was not automatically useful unless the interface made those controls understandable.

5.7.3 AdaptaFace expanded participants' usable gesture vocabulary. Participants treated extension as more than a one-time substitution. EP8 said, *"I like to switch expressions when I get tired of one,"* while EP13 contrasted selectable mode's eight choices with custom mode, where *"I can just do whatever I want."* These accounts reveal a tradeoff across modes: selectable gestures offered a bounded set that could reduce setup decisions, while custom gestures improved personal fit and memorability. Participants valued small repertoires that could accommodate changes in fatigue, task, or context rather than one permanently optimal gesture. Preferences for subtle gestures in public also linked vocabulary extension to social comfort. Together, these findings suggest that accessible personalization should support families of alternatives and allow mappings to evolve with the user's situation.

6 Offline Validation: Cross-Dataset, Cross-Model, and Model-Transfer Analyses

We next examine whether the compatibility-layer mechanism shows signs of transfer beyond the exact user-study deployment setting. We conducted offline validation on Aff-Wild2, a large in-the-wild facial-behavior dataset with AU annotations [20–22, 40]. This dataset provides diverse, unconstrained facial behavior at scale while covering the shared AUs used in our system, making it suitable for controlled validation of the underlying assistance mechanism.

The three settings progressively relax alignment between perturbation generation and evaluation. In *cross-dataset validation*, perturbations were generated with OpenFace and evaluated by OpenFace on Aff-Wild2. In *cross-model validation*, the same validation logic was instantiated with OpenGraphAU, i.e., perturbations were generated and evaluated within a different recognizer family on the same dataset. In *model-transfer validation*, perturbations generated with OpenGraphAU were evaluated by OpenFace, testing whether the input-side effect transferred across recognizers.

Following the logic of AdaptaFace, we focus on two state types. *Enhancement* tests whether assistance can strengthen recognition of a target AU that should become active. *Substitution* tests whether assistance can steer recognition away from a present source AU toward an absent target AU. For each setting, we compared Baseline, adversarial assistance, and random perturbations. Table 7 reports the aggregate baseline and adversarial-assistance results, while the random-perturbation condition remained close to baseline across settings and is summarized in the text below.

Across both state types, cross-dataset validation showed the strongest aggregate evidence of successful steering. In the enhancement case, target-confidence success increased from **0.3238** to **0.7591**, and target top-1 from **0.3853** to **0.7863**. In the substitution case, target-confidence success increased from **0.0223** to **0.5248**, and target top-1 from **0.0401** to **0.5917**. These gains suggest that the compatibility-layer mechanism can remain effective under dataset shift when perturbation generation and evaluation are still aligned within the same recognizer.

Cross-model validation with OpenGraphAU showed the same overall direction, indicating that the mechanism is not tied to OpenFace alone. In enhancement, target-confidence success increased from **0.3692** to **0.7175**, and target top-1 from **0.4553** to **0.7179**. In substitution, target-confidence success increased from **0.0087** to **0.4378**, and target top-1 from **0.0174** to **0.4387**. At the same time, the pair-level results were more target-dependent than in the cross-dataset setting: some AU pairs approached near-saturation under assistance, while others remained comparatively difficult.

By contrast, model-transfer validation from OpenGraphAU to OpenFace was weaker. In enhancement, target-confidence success increased only from **0.2407** to **0.2665**, and target top-1 from **0.3022** to **0.3475**. In substitution, target-confidence success increased from **0.0368** to **0.0432**, and target top-1 from **0.0584** to **0.0812**. These modest aggregate gains suggest that transfer across recognizers is possible in some cases, but remains less reliable than settings in which perturbation generation and evaluation are aligned.

Across all settings, random perturbations remained close to baseline and consistently below adversarial assistance. This indicates that the observed gains were not explained by perturbation magnitude alone, but depended on targeted steering toward the intended AU.

These results clarify the evaluated boundary of AdaptaFace. Input-level assistance transfers most strongly when the perturbation generator and evaluator are aligned, remains viable when instantiated with a different recognizer family, and becomes substantially less reliable under cross-recognizer transfer.

7 Discussion

7.1 Personalization as Accessibility Mediation

AdaptaFace reframes adversarial perturbation from a robustness threat into an accessibility mechanism. Rather than asking users to make their movements conform to a canonical vocabulary, it mediates the recognizer-facing image so that feasible movements can become usable commands. This complements ability-based design, which asks systems to adapt to users' abilities [38], and ability-mediating design, which foregrounds how existing sensorimotor abilities can be mediated to enable new interaction abilities [34].

Personalization is therefore the central contribution. Enhancement modes improve use of the recognizer's existing vocabulary, while extension modes let participants use subtler, more memorable, or personally meaningful movements without retraining the recognizer. The qualitative findings also resist a single optimal mode: selectable gestures can reduce setup decisions, custom gestures can improve fit and memorability, and alternatives matter when fatigue or context changes. Adversarial assistance carries these personalized bindings through an unchanged image-recognition pathway; the evaluation estimates the complete pipeline rather than a separately isolated perturbation effect.

7.2 Integration and Why Image Mediation Matters

AdaptaFace is intended for systems in which a developer can intercept camera frames before recognition but cannot retrain or replace the downstream recognizer or its command mapping. Integration requires access to that image stream, a differentiable proxy with a compatible target space, and a registration layer mapping user-compatible gestures to internal targets. Proxy/recognizer pairings should be validated rather than assumed to transfer, and facial images and personalized templates should remain local where possible.

Trajectory matching alone selects an internal target; without perturbation, the recognizer still receives the raw frame and therefore follows the same recognizer-facing path as Baseline. Directly emitting the matched label as a command is a valid alternative when an application exposes an appropriate API, but it bypasses the image recognizer and requires a new event source. Image mediation addresses the complementary setting in which the existing image recognizer is the available integration boundary. The choice depends on application access, recognizer coupling, privacy, latency, and the feedback users need to inspect and revise mappings.

Figure 3: Examples from the three offline validation settings: cross-dataset (left), cross-model (middle), and model transfer (right). Each triplet shows the original input, the assisted image, and a normalized visualization of the perturbation. Numbers below the original and assisted images indicate target confidence before and after assistance.

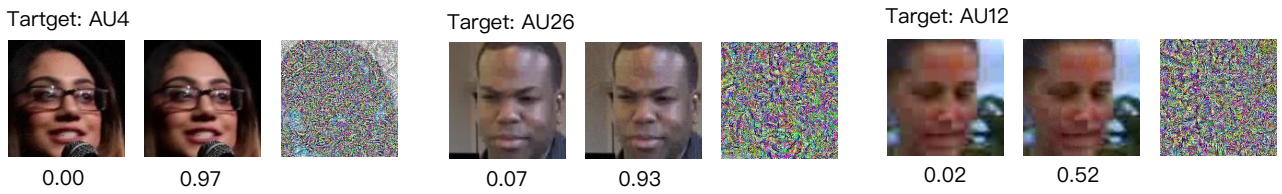

Table 7: Aggregate offline validation results across three settings: cross-dataset, cross-model, and model-transfer. S = substitution (target= 0) tests whether assistance can steer recognition toward a target AU that is initially absent, whereas E = enhancement (target= 1) tests whether assistance can strengthen recognition of a target AU that should become active. Conf. reports target-confidence success rate, and Top-1 reports the proportion of examples for which the target AU became the top-ranked prediction. Gains are computed relative to the baseline condition.

Setting	Case	Base Conf.	Adv Conf.	Conf. Gain	Base Top-1	Adv Top-1	Top-1 Gain
Cross-dataset	Substitution	0.0223	0.5248	0.5026	0.0401	0.5917	0.5516
Cross-dataset	Enhancement	0.3238	0.7591	0.4352	0.3853	0.7863	0.4011
Cross-model	Substitution	0.0087	0.4378	0.4291	0.0174	0.4387	0.4212
Cross-model	Enhancement	0.3692	0.7175	0.3483	0.4553	0.7179	0.2626
Model-transfer	Substitution	0.0368	0.0432	0.0064	0.0584	0.0812	0.0228
Model-transfer	Enhancement	0.2407	0.2665	0.0258	0.3022	0.3475	0.0452

These requirements are not specific to facial input. Camera-based hand or body gesture recognizers could use the same pattern when developers can intercept the input stream and construct a proxy with a compatible target space. Beyond vision, one possible direction is fixed class-based recognition: for example, a modality-specific mediation layer could map user-compatible vocalizations to commands supported by a closed-set spoken-command recognizer. Extending the approach would require modality-specific optimization and perceptual constraints, temporal matching, and validation across recognizers. For new user populations, registration and feedback should be designed around their abilities and preferences rather than reuse the facial gesture repertoire evaluated here.

7.3 Scope, Limitations, and Future Work

The online study used the same OpenFace-based AU model for perturbation generation and post-assistance scoring, so its gains characterize an aligned proxy/evaluator setting. Offline validation showed that the mechanism could also be instantiated with OpenGraphAU, but OpenGraphAU-to-OpenFace transfer was substantially weaker. Targeted transfer is known to be difficult, although ensemble, feature-space, and intermediate-level objectives offer possible improvement strategies [18, 23, 24, 36]. These results define a current boundary rather than evidence of universal black-box portability.

The study compared the complete Baseline and AdaptaFace pipelines and did not isolate matching, target selection, and perturbation generation through a component ablation. Its remote deployment also did not systematically manipulate pose, lighting, occlusion, webcam quality, or within-person change. Future work should benchmark matching and mediation across those conditions and recognizer pairings, examine longer-term changes in fatigue and gesture use, and compare image mediation with direct triggering where both architectures are available.

8 Conclusion

We presented AdaptaFace, a facial interaction system that treats personalization as input-side accessibility mediation. The complete pipeline improved task performance, perceived control, usability, and workload relative to Baseline, while extension modes broadened participants' usable gesture vocabularies. Together, the results show how feasible, memorable movements can be carried through an unchanged AU-based recognition pathway without retraining. The present evidence remains bounded to proxy-compatible target spaces and does not isolate every pipeline component, but it establishes input-side mediation as a promising direction for more user-compatible facial interaction.

Acknowledgments

We thank all participants for their time and valuable contributions, and the University of Bristol for supporting this research.

References

[1] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. 2016. OpenFace: An Open Source Facial Behavior Analysis Toolkit. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 1–10. https://doi.org/10.1109/WACV.2016.7477553

[2] R. Bates and H. O. Istance. 2003. Why are eye mice unpopular? A detailed comparison of head and eye controlled assistive technology pointing devices. *Universal Access in the Information Society* 2, 3 (2003), 280–290. https://doi.org/10.1007/s10209-003-0053-y

[3] Margrit Betke, James Gips, and Peter Fleming. 2002. The Camera Mouse: Visual Tracking of Body Features to Provide Computer Access for People with Severe Disabilities. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 10, 1 (2002), 1–10. https://doi.org/10.1109/TNSRE.2002.1021581

[4] Tom B. Brown, Dandelion Mané, Aurko Roy, Martin Abadi, and Justin Gilmer. 2017. Adversarial Patch. *arXiv preprint arXiv:1712.09665* (2017). https://arxiv.org/abs/1712.09665

[5] Sasha Costanza-Chock. 2020. *Design Justice: Community-Led Practices to Build the Worlds We Need.* MIT Press.

[6] Brendan David-John, Kevin R. B. Butler, and Eakta Jain. 2022. For Your Eyes Only: Privacy-preserving Eye-tracking Datasets. In *Proceedings of the 2022 ACM Symposium on Eye Tracking Research & Applications (ETRA)*. 1–6. https://doi.org/10.1145/3517031.3529618

[7] Gamaleldin F. Elsayed, Ian Goodfellow, and Jascha Sohl-Dickstein. 2018. Adversarial Reprogramming of Neural Networks. *arXiv preprint arXiv:1806.11146* (2018). https://arxiv.org/abs/1806.11146

[8] Mingming Fan, Zhen Li, and Franklin Mingzhe Li. 2020. Eyelid Gestures on Mobile Devices for People with Motor Impairments. In *The 22nd International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '20)*. https://doi.org/10.1145/3373625.3416987

[9] Ruth C. Fong and Andrea Vedaldi. 2017. Interpretable Explanations of Black Boxes by Meaningful Perturbation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 3429–3437. https://openaccess.thecvf.com/content_ICCV_2017/papers/Fong_Interpretable_Explanations_of_ICCV_2017_paper.pdf

[10] Krzysztof Z. Gajos, Daniel S. Weld, and Jacob O. Wobbrock. 2010. Automatically Generating Personalized User Interfaces with Supple. *Artificial Intelligence* 174, 12–13 (2010), 910–950. https://doi.org/10.1016/j.artint.2010.05.005

[11] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. *arXiv preprint arXiv:1412.6572* (2015). https://arxiv.org/abs/1412.6572

[12] Google AI Edge. 2026. Face Landmarker Guide. https://ai.google.dev/edge/mediapipe/solutions/vision/face_landmarker. Accessed 2026-08-01.

[13] Kristen Grauman, Margrit Betke, John Lombardi, James Gips, and Gary R. Bradski. 2003. Communication via eye blinks and eyebrow raises: video-based human-computer interfaces. *Universal Access in the Information Society* 2, 4 (2003), 359–373. https://doi.org/10.1007/s10209-003-0062-x

[14] Ivan Grishchenko, Artsiom Ablavatski, Yury Kartynnik, Karthik Raveendran, and Matthias Grundmann. 2020. Attention Mesh: High-fidelity Face Mesh Prediction in Real-time. *arXiv preprint arXiv:2006.10962* (2020). https://arxiv.org/abs/2006.10962

[15] Junfeng He, Khoi Pham, Nachiappan Valliappan, Pingmei Xu, Chase Roberts, Dmitry Lagun, and Vidhya Navalpakkam. 2019. On-Device Few-Shot Personalization for Real-Time Gaze Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*. 1149–1158. https://doi.org/10.1109/ICCVW.2019.00146

[16] Jiewen Hu, Leena Mathur, Paul Pu Liang, and Louis-Philippe Morency. 2025. OpenFace 3.0: A Lightweight Multitask System for Comprehensive Facial Behavior Analysis. *arXiv preprint arXiv:2506.02891* (2025).

[17] Xueliang Huo and Maysam Ghovanloo. 2009. Using Unconstrained Tongue Motion as an Alternative Control Mechanism for Wheeled Mobility. *IEEE Transactions on Biomedical Engineering* 56, 6 (2009), 1719–1726. https://doi.org/10.1109/TBME.2009.2018632

[18] Nathan Inkawhich, Wei Wen, Hai Li, and Yiran Chen. 2019. Feature Space Perturbations Yield More Transferable Adversarial Examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[19] J. Kim, H. Park, J. Bruce, D. Rowles, J. Holbrook, B. Nardone, S. Skaar, and M. Ghovanloo. 2013. The Tongue Enables Computer and Wheelchair Control for People with Spinal Cord Injury. *PLOS ONE* 8, 11 (2013), e80933. https://doi.org/10.1371/journal.pone.0080933

[20] Dimitrios Kollias, Panagiotis Tzirakis, Alice Baird, Alan Cowen, and Stefanos Zafeiriou. 2023. Abaw: Valence-arousal estimation, expression recognition, action unit detection & emotional reaction intensity estimation challenges. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5888–5897.

[21] Dimitrios Kollias, Panagiotis Tzirakis, Mihalis A Nicolaou, Athanasios Papaioannou, Guoying Zhao, Björn Schuller, Irene Kotsia, and Stefanos Zafeiriou. 2019. Deep affect prediction in-the-wild: Aff-wild database and challenge, deep architectures, and beyond. *International Journal of Computer Vision* (2019), 1–23.

[22] Dimitrios Kollias and Stefanos Zafeiriou. 2019. Expression, Affect, Action Unit Recognition: Aff-Wild2, Multi-Task Learning and ArcFace. *arXiv preprint arXiv:1910.04855* (2019).

[23] Qing Li et al. 2023. Improving Adversarial Transferability via Intermediate-level Perturbation Decay. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[24] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. 2017. Delving into Transferable Adversarial Examples and Black-box Attacks. In *International Conference on Learning Representations*.

[25] Cheng Luo, Siyang Song, Weicheng Xie, Linlin Shen, and Hatice Gunes. 2022. Learning Multi-dimensional Edge Feature-based AU Relation Graph for Facial Action Unit Recognition. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. 1239–1246. https://doi.org/10.24963/ijcai.2022/173

[26] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/pdf?id=rJzIBfZAb

[27] Päivi Majaranta and Andreas Bulling. 2014. Eye Tracking and Eye-Based Human–Computer Interaction. In *Advances in Physiological Computing*, Stephen H. Fairclough and Kiel Gilleade (Eds.). Springer, 39–65. https://doi.org/10.1007/978-1-4471-6392-3_3

[28] Omar Namnakani, Yasmeen Abdrabou, Jonathan Grizou, Afonso Esteves, and Mohamed Khamis. 2023. Comparing Dwell Time, Pursuits, and Gaze Gestures for Target Selection on Mobile Phones While Sitting and Walking. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3544548.3580871

[29] Seonwook Park, Shalini De Mello, Pavlo Molchanov, Umar Iqbal, Otmar Hilliges, and Jan Kautz. 2019. Few-Shot Adaptive Gaze Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. 9368–9377. https://openaccess.thecvf.com/content_ICCV_2019/papers/Park_Few-Shot_Adaptive_Gaze_Estimation_ICCV_2019_paper.pdf

[30] Soroush Shahi, Vimal Mollyn, Cori Tymoszek Park, Richard Kang, Asaf Liberman, Oron Levy, Jun Gong, Abdelkareem Bedri, and Gierad Laput. 2024. Vision-Based Hand Gesture Customization from a Single Demonstration. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24)*. https://doi.org/10.1145/3654777.3676378

[31] Julian Steil, Inken Hagestedt, Michael Xuelin Huang, and Andreas Bulling. 2019. Privacy-aware Eye Tracking Using Differential Privacy. In *Proceedings of the 2019 ACM Symposium on Eye Tracking Research & Applications (ETRA)*. 1–9. https://doi.org/10.1145/3314111.3319915

[32] Qin Sun, Yunqi Hu, Mingming Fan, Jingting Li, and Su-Jing Wang. 2024. "Can It Be Customized According to My Motor Abilities?": Toward Designing User-Defined Head Gestures for People with Dystonia. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. https://doi.org/10.1145/3613904.3642378

[33] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing Properties of Neural Networks. *arXiv preprint arXiv:1312.6199* (2013). https://arxiv.org/abs/1312.6199

[34] Radu-Daniel Vatavu. 2022. Sensorimotor Realities: Formalizing Ability-Mediating Design for Computer-Mediated Reality Environments. In *2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 685–694. https://doi.org/10.1109/ISMAR55827.2022.00086

[35] Mike Wald. 2021. AI Data-Driven Personalisation and Disability Inclusion. *Informatics* 8, 1 (2021), 10. https://doi.org/10.3390/informatics8010010

[36] Xiaosen Wang, Xuanran He, Jingdong Wang, and Kun He. 2021. Admix: Enhancing the Transferability of Adversarial Attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[37] Johann Wentzel, Alessandra Luz, Martez E. Mott, and Daniel Vogel. 2025. MotionBlocks: Modular Geometric Motion Remapping for More Accessible Upper Body Movement in Virtual Reality. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. Article 608, 16 pages. https://doi.org/10.1145/3706598.3713837

[38] Jacob O. Wobbrock, Shaun K. Kane, Krzysztof Z. Gajos, Susumu Harada, and Jon Froehlich. 2011. Ability-Based Design: Concept, Principles and Examples. *ACM Transactions on Accessible Computing (TACCESS)* 3, 3, Article 9 (2011), 27 pages. https://doi.org/10.1145/1952383.1952384

[39] Momona Yamagami, Alexandra A. Portnova-Fahreeva, Junhan Kong, Jacob O. Wobbrock, and Jennifer Mankoff. 2023. How Do People with Limited Movement Personalize Upper-Body Gestures? Considerations for the Design of Personalized and Accessible Gesture Interfaces. In *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '23)*. https://doi.org/10.1145/3597638.3608430

[40] Stefanos Zafeiriou, Dimitrios Kollias, Mihalis A Nicolaou, Athanasios Papaioannou, Guoying Zhao, and Irene Kotsia. 2017. Aff-wild: Valence and arousal 'in-the-wild' challenge. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE, 1980–1987.

[41] Siyu Zhang, Yelu Gu, Kirsten Cater, and Oussama Metatla. 2026. Beyond Accuracy: Auditing Allocative Harms in Facial-Gesture Recognition for People with Motor Impairments. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems* (Barcelona, Spain) *(CHI '26)*. https://doi.org/10.1145/3772318.3791927

[42] Xuan Zhao, Mingming Fan, and Teng Han. 2022. "I Don't Want People to Look At Me Differently": Designing User-Defined Above-the-Neck Gestures for People with Upper Body Motor Impairments. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3491102.3517552

A AdaptaFace Implementation Details

A.1 PGD-style Perturbation Optimization

AdaptaFace generates adversarial assistance using a PGD-style targeted optimization procedure with a random start. Given an input frame x and a target recognizer output y^*, the system iteratively updates a perturbation δ to reduce the proxy-model loss for y^* while constraining the perturbation magnitude within a bounded L_∞ norm ball. This yields a visually subtle but effective modification of the input that steers recognition toward the intended target.

At each iteration, AdaptaFace computes the gradient of the targeted loss with respect to the current perturbed input, updates the perturbation in the loss-reducing direction, and projects the result back into the feasible perturbation set. We use only a small number of iterations in practice to preserve real-time performance. Panel E of Figure 1 shows a representative perturbation example; for visibility, the perturbation visualization is magnified.

Algorithm 1 AdaptaFace perturbation engine (per frame)

Require: frame x, target output y^*, proxy model f_{proxy}, step size α, bound ϵ, iterations K

$\delta \leftarrow \text{clip}_{[0,1]}(x+\eta) - x$, where $\eta \sim \mathcal{U}([-\epsilon, \epsilon])$
for $k = 1$ to K **do**
  $g \leftarrow \nabla_x \mathcal{L}(f_{\text{proxy}}(x+\delta), y^*)$
  $\delta \leftarrow \delta - \alpha \cdot \text{sign}(g)$
  $\delta \leftarrow \text{Proj}_{\|\delta\|_\infty \le \epsilon,\, x+\delta \in [0,1]}(\delta)$
end for
return $x + \delta$

A.2 Target Selection in Matched Modes

In the matched modes, AdaptaFace assigns a target recognizer output during registration rather than during runtime interaction. Let $\mathcal{G} = \{g_1, \ldots, g_N\}$ denote the candidate set. In Matched Enhancement, $\mathcal{G}$ corresponds to the set of interface-supported gestures. In Matched Extension, $\mathcal{G}$ corresponds to the set of available recognizer outputs considered as candidate mediation targets for user-defined gestures.

Given a registered gesture sample or sequence X^{reg}, AdaptaFace performs registration-time candidate evaluation by estimating how well the assisted registration input aligns with each candidate $g_i \in \mathcal{G}$. Let $c_i(X^{\text{reg}})$ denote the downstream recognizer's confidence score for candidate g_i. The matched target is selected as

$$g^* = \arg\max_{g_i \in \mathcal{G}} c_i(X^{\text{reg}}). \tag{2}$$

The selected target is accepted only if its confidence exceeds a minimum threshold τ_c:

$$c_{g^*}(X^{\text{reg}}) \ge \tau_c. \tag{3}$$

Optionally, AdaptaFace also applies a confidence-margin criterion:

$$c_{g^*}(X^{\text{reg}}) - c_{g^{(2)}}(X^{\text{reg}}) \ge \tau_m, \tag{4}$$

where $g^{(2)}$ is the second-highest-scoring candidate. If these criteria are not satisfied, no matched target is assigned.

To improve robustness during registration, AdaptaFace can aggregate confidence scores across multiple frames in the recorded sample or require the same candidate to remain top-ranked across a short temporal window before confirming the match.

A.3 Runtime Gesture Detection

Before applying assistance, AdaptaFace first determines whether the user's ongoing facial motion corresponds to one of the registered gestures. Let $\mathcal{R} = \{r_1, \ldots, r_M\}$ denote the set of registered gesture trajectories. Each registered gesture r_j is associated with a target recognizer output determined during registration, either predefined in the fixed modes or selected through matching in the matched modes. Each incoming frame is represented by an unnormalized eight-dimensional AU activation vector $\mathbf{a}_t$. At runtime, the system maintains a short causal sliding window

$$A_t = \{\mathbf{a}_{t-W+1}, \ldots, \mathbf{a}_t\}, \tag{5}$$

where W is the window length. Writing the aligned stored trajectory as $r_j = \{\mathbf{r}_{j,1}, \ldots, \mathbf{r}_{j,W}\}$, the matcher computes

$$D_j(A_t, r_j) = \frac{1}{W} \sum_{i=1}^{W} \left\| \mathbf{a}_{t-W+i} - \mathbf{r}_{j,i} \right\|_2, \tag{6}$$

followed by the distance-derived similarity

$$s_j(A_t, r_j) = \frac{1}{1 + D_j(A_t, r_j)}. \tag{7}$$

The best-matching registered gesture is defined as

$$r^* = \arg\max_{r_j \in \mathcal{R}} s_j(A_t, r_j), \tag{8}$$

with equal scores resolved deterministically by template identifier.

A gesture is considered detected only if the best match exceeds a minimum similarity threshold:

$$s_{r^*}(A_t, r^*) \ge \tau_{\text{match}}. \tag{9}$$

The matcher exposes T as a configurable stability requirement and can additionally require the same gesture to remain top-ranked and above threshold for T consecutive windows before confirmation. The study configuration used $W = 1$, $\tau_{\text{match}} = 2/3$, and $T = 3$, with no AU-vector normalization. The similarity threshold is equivalent to a Euclidean-distance threshold of 0.5. Thus, confirmation required three consecutive above-threshold frame-level matches to the same template, suppressing transient matches.

Rather than waiting for a full gesture sequence to complete, AdaptaFace uses short-window matching to detect gesture onset online. Once a registered gesture is detected, the compatibility layer applies adversarial assistance frame by frame to subsequent input during interaction, steering recognition toward the target previously assigned to that registered gesture.

B Extended User-Study Materials

B.1 Statistical Model Specifications

B.1.1 Modeling Setup. We analyzed repeated-measures quantitative data using two complementary model families. CR was summarized as the within-block proportion of required action repetitions completed. For inferential analysis, full-block completion, a binary outcome indicating whether all required repetitions in a task block were completed, was analyzed using a binomial generalized estimating equation (GEE) with a logit link and participant-level clustering. Continuous repeated-measures outcomes were analyzed using linear mixed-effects models (LMMs) with participant-level random intercepts. These outcomes included task completion time (TCT), success-normalized task time (SNT), ease of use, and perceived control.

Technology (Baseline vs. AdaptaFace), Task, and Personalization Mode were entered as fixed effects in the repeated-measures models. Because no Technology × Task or Technology × Mode interaction reached significance in the final analyses, the main paper focuses on Technology main effects, while task- and mode-level patterns are interpreted descriptively. System-level SUS and NASA Raw TLX were collected once per technology condition and are therefore summarized descriptively in the main paper rather than modeled as repeated-measures outcomes.

B.1.2 Objective Performance Models.

Full-Block Completion. The binary full-block completion indicator derived from CR was modeled using a binomial GEE with a logit link:

$$\Pr(\mathrm{FBC}_{ij} = 1) = \mathrm{logit}^{-1}\left(\beta_0 + \beta_T \mathrm{Technology}_{ij} + \beta_{Ta}\mathrm{Task}_{ij} + \beta_M \mathrm{Mode}_{ij}\right),$$

where i indexes participants and j indexes task blocks.

Task Completion Time (TCT).. Task completion time was modeled using a linear mixed-effects model:

$$\mathrm{TCT}_{ij} = \beta_0 + \beta_T \mathrm{Technology}_{ij} + \beta_{Ta}\mathrm{Task}_{ij} + \beta_M \mathrm{Mode}_{ij} + u_i + \epsilon_{ij},$$

where u_i is a participant-level random intercept.

Success-Normalized Task Time (SNT).. Success-normalized task time was modeled analogously:

$$\mathrm{SNT}_{ij} = \beta_0 + \beta_T \mathrm{Technology}_{ij} + \beta_{Ta}\mathrm{Task}_{ij} + \beta_M \mathrm{Mode}_{ij} + u_i + \epsilon_{ij}.$$

B.1.3 Subjective Rating Models.

Ease and Perceived Control. Ease and perceived control ratings were analyzed using linear mixed-effects models with the same fixed- and random-effects structure:

$$Y_{ij} = \beta_0 + \beta_T \mathrm{Technology}_{ij} + \beta_{Ta}\mathrm{Task}_{ij} + \beta_M \mathrm{Mode}_{ij} + u_i + \epsilon_{ij},$$

where Y_{ij} denotes either Ease or Control.

System-Level SUS and NASA Raw TLX.. SUS and NASA Raw TLX were collected once per participant for each technology condition rather than once per task block. For this reason, they are treated as system-level comparisons in the main paper rather than repeated-measures model outcomes.

B.1.4 Fixed-Effect Tables. Table 8 reports the GEE fixed effects for full-block completion. Tables 9 and 10 report the LMM fixed effects for TCT and SNT. Tables 11 and 12 report the LMM fixed effects for Ease and Control. Across these tables, the reference levels are Baseline, Click, and Fixed Extension.

B.2 Formative-Study Interview Procedure

The main paper reports the interview sequence and analysis purpose in Section 3. Sessions were conducted remotely by video conference, recorded with consent, and structured to preserve participants' ability to pause or withdraw.

B.3 Extended Design Principles from the Formative Study

Based on the formative findings, we derived six design principles for inclusive facial-gesture personalization. In the main paper, we summarize these principles briefly; here we provide a slightly fuller description of how each informed the design of AdaptaFace.

DP1. Support user-compatible gestures rather than only canonical templates. Personalization should accommodate gestures that users can comfortably and reliably perform, even when these differ from recognizer-defined expressions. This follows from participants' accounts that subtle, asymmetric, or personally habitual movements were often more feasible than standardized facial gestures.

DP2. Minimize user burden during personalization. Personalization should reduce repeated demonstration, calibration, and setup effort, especially for users whose movement quality varies across posture, fatigue, or day-to-day condition.

DP3. Enable user-defined gesture vocabularies. Users should be able to bind their own feasible and meaningful facial actions to system commands, rather than selecting only from a closed predefined set. This supports interaction that reflects users' existing expressive habits instead of forcing all input into a fixed recognizer vocabulary.

DP4. Preserve user control over personalization. Assistance should remain legible and user-governed, allowing users to understand and shape how their gestures are interpreted. Personalization should therefore support explicit gesture–command mappings and avoid opaque adaptation that removes user agency.

DP5. Preserve compatibility with existing recognition pipelines. To lower deployment barriers, personalization should work with off-the-shelf recognizers rather than requiring wholesale replacement or retraining of downstream models. This also supports broader applicability across heterogeneous recognition pipelines, rather than binding personalization to a single recognizer architecture.

DP6. Support low-effort and socially acceptable interaction. The system should accommodate gestures that are low-fatigue, low-amplitude, and appropriate for everyday settings, including public or shared environments. Inclusive personalization should therefore optimize not only for recognizability, but also for comfort, sustainability, and social acceptability.

Table 8: GEE fixed effects for full-block completion (logit link). Only interpretable fixed-effect terms are reported.

Term	Estimate	StdErr	z	p	CI_{low}	CI_{high}
Task: Swipe	-0.326	0.132	-2.461	0.014	-0.585	-0.066
Task: Zoom In	-0.346	0.131	-2.637	0.008	-0.604	-0.089
Task: Zoom Out	-0.066	0.135	-0.493	0.622	-0.330	0.198
Mode: Matched Extension	0.588	0.133	4.410	< .001	0.327	0.849
Mode: Fixed Enhancement	1.230	0.133	9.242	< .001	0.970	1.491
Mode: Matched Enhancement	1.153	0.133	8.697	< .001	0.893	1.413
Technology: AdaptaFace	1.116	0.094	11.864	< .001	0.931	1.300

Table 9: LMM fixed effects for Task Completion Time (seconds). Only interpretable fixed-effect terms are reported.

Term	Estimate	StdErr	z	p	CI_{low}	CI_{high}
Technology: AdaptaFace	-3.183	0.402	-7.924	< .001	-3.971	-2.395
Task: Swipe	0.854	0.655	1.304	0.192	-0.431	2.138
Task: Zoom In	0.827	0.636	1.300	0.194	-0.420	2.074
Task: Zoom Out	0.399	0.665	0.600	0.549	-0.905	1.704
Mode: Matched Extension	-3.879	0.641	-6.052	< .001	-5.136	-2.622
Mode: Fixed Enhancement	-0.990	0.657	-1.507	0.132	-2.278	0.299
Mode: Matched Enhancement	-1.453	0.659	-2.206	0.027	-2.744	-0.163

Table 10: LMM fixed effects for Success-Normalized Task Time (seconds per successful action). Only interpretable fixed-effect terms are reported.

Term	Estimate	StdErr	z	p	CI_{low}	CI_{high}
Technology: AdaptaFace	-12.157	0.963	-12.626	< .001	-14.046	-10.269
Task: Swipe	-0.420	1.464	-0.287	0.774	-3.289	2.448
Task: Zoom In	-6.441	1.445	-4.459	< .001	-9.272	-3.610
Task: Zoom Out	5.563	1.526	3.646	< .001	2.571	8.556
Mode: Matched Extension	-10.378	1.456	-7.126	< .001	-13.232	-7.525
Mode: Fixed Enhancement	-7.334	1.488	-4.933	< .001	-10.250	-4.418
Mode: Matched Enhancement	-5.900	1.490	-3.959	< .001	-8.830	-2.970

Table 11: LMM fixed effects for Ease ratings (1–5). Only interpretable fixed-effect terms are reported.

Term	Estimate	StdErr	z	p	CI_{low}	CI_{high}
Technology: AdaptaFace	0.674	0.051	13.118	< .001	0.573	0.775
Task: Swipe	0.056	0.050	1.115	0.265	-0.042	0.154
Task: Zoom In	-0.061	0.049	-1.241	0.215	-0.158	0.036
Task: Zoom Out	0.145	0.051	2.859	0.004	0.045	0.245
Mode: Matched Extension	0.529	0.051	10.377	< .001	0.428	0.629
Mode: Fixed Enhancement	-0.093	0.052	-1.788	0.074	-0.195	0.009
Mode: Matched Enhancement	-0.062	0.052	-1.192	0.233	-0.164	0.039

These principles point toward a different locus of personalization: a mediating mechanism at the input level that preserves users' own expressive habits while helping existing recognizers interpret them more reliably.

B.4 Remote Study Setup

To illustrate the evaluation context, Figure 4 shows representative remote-study setups.

B.5 Task-Level Performance

Table 13 provides the task-level performance breakdown for the four interaction tasks.

B.6 Ease and Control Ratings

Table 14 reports subjective ease and control ratings overall, by task, and by personalization mode.

Table 12: LMM fixed effects for Perceived Control ratings (1–5). Only interpretable fixed-effect terms are reported.

Term	Estimate	StdErr	z	p	CI_{low}	CI_{high}
Technology: AdaptaFace	1.124	0.055	20.478	< .001	1.016	1.231
Task: Swipe	0.045	0.054	0.824	0.410	-0.061	0.152
Task: Zoom In	-0.006	0.054	-0.108	0.914	-0.111	0.100
Task: Zoom Out	0.423	0.056	7.528	< .001	0.313	0.533
Mode: Matched Extension	0.503	0.055	9.119	< .001	0.395	0.612
Mode: Fixed Enhancement	-0.331	0.056	-5.913	< .001	-0.440	-0.222
Mode: Matched Enhancement	-0.197	0.056	-3.548	< .001	-0.307	-0.087

Figure 4: Representative remote-study setups. Participants completed the evaluation using their own devices and webcams in varied postures and home environments.

Table 13: Task-level performance breakdown (mean (SD)). Zoom tasks are reported separately.

Task	CR %		TCT s		SNT s		Conf.	
	Baseline	AdaptaFace	Baseline	AdaptaFace	Baseline	AdaptaFace	Baseline	AdaptaFace
Click	70.0 (41.1)	95.5 (18.2)	11.5 (7.4)	8.4 (4.7)	17.6 (19.2)	8.8 (5.0)	0.78 (0.37)	0.92 (0.24)
Swipe	55.5 (46.9)	87.0 (29.0)	17.6 (12.0)	13.3 (8.7)	21.8 (23.8)	18.2 (24.4)	0.56 (0.45)	0.87 (0.30)
Zoom In	52.1 (43.9)	89.1 (28.1)	24.0 (27.1)	18.3 (10.2)	96.1 (292.9)	20.7 (11.9)	0.62 (0.42)	0.88 (0.28)
Zoom Out	61.1 (45.6)	90.3 (25.6)	13.1 (9.2)	13.4 (8.4)	14.2 (14.2)	16.3 (14.5)	0.64 (0.45)	0.87 (0.28)

Table 14: Ease and control ratings (1–5; higher is better).

(a) Overall

	Ease	Control
Baseline	3.70	3.15
AdaptaFace	4.35	4.27

(b) By task

Task	Ease		Control	
	Base	Ada	Base	Ada
Click	3.55	4.13	2.82	3.99
Swipe	3.66	4.39	3.11	4.26
Zoom In	3.62	4.34	3.05	4.27
Zoom Out	3.78	4.38	3.63	4.53

(c) By mode

Mode	Ease		Control	
	Base	Ada	Base	Ada
Fixed Enhancement	3.58	4.22	2.96	4.04
Matched Enhancement	3.62	4.28	2.98	4.14
Fixed Extension	3.75	4.41	3.22	4.33
Matched Extension	3.86	4.51	3.44	4.55

B.7 Gesture Vocabulary Inventory

Table 15 summarizes the user-defined gestures introduced in the extension-oriented modes.

C Additional Offline Validation Examples

Table 15: Gesture-level breakdown of user-defined gestures introduced in the extension-oriented modes, with descriptive completion rates by technology and improvement (Δ).

Action (English)	Interactions	Participants	Count	Baseline (%)	AdaptaFace (%)	Δ (%)
Right Brow Raise	Swipe	EP2	1	60.0	100.0	40.0
Eye Widen	Zoom In	EP16	1	60.0	100.0	40.0
Half-squinted Eyes	Zoom Out	EP13	1	0.0	100.0	100.0
Close Eyes	Swipe, Click, Zoom In	EP12, EP16, EP3, EP5, EP8	11	14.5	66.0	51.5
Close Left Eye	Swipe, Zoom In/Out	EP10, EP14, EP5, EP9	7	22.9	90.0	67.1
Close Right Eye	Swipe, Click, Zoom In/Out	EP10, EP11, EP14, EP2, EP3, EP6, EP8	11	40.0	83.6	43.6
Eye Look Left	Swipe	EP9	1	0.0	100.0	100.0
Eye Look Up	Click, Zoom Out	EP12, EP13, EP6	4	25.0	75.0	50.0
Eye Look Down	Zoom Out	EP13	1	40.0	100.0	60.0
Puffing up cheeks	Zoom In/Out	EP13, EP2, EP3, EP6	5	20.0	100.0	80.0
Mouth Widen	Click, Zoom In	EP1, EP10, EP13, EP8, EP15	6	38.3	100.0	61.7
Lip Press	Swipe, Click, Zoom In/Out	EP13, EP9, EP2, EP3, EP6	6	33.3	96.7	63.4
Pout	Swipe, Click, Zoom In	EP1, EP11, EP13, EP15, EP16, EP3, EP8, EP9	13	33.8	91.7	57.9
Stretch Left Lip	Zoom In	EP7	1	0.0	60.0	60.0
Stretch Right Lip	Zoom In	EP7	1	20.0	100.0	80.0
Raise Left Lip Corner	Swipe, Click	EP11, EP3, EP9	5	35.0	88.0	53.0
Raise Right Lip Corner	Click	EP14, EP9, EP2	4	15.0	70.0	55.0
Smile (teeth)	Swipe	EP1, EP13, EP4	3	26.7	86.7	60.0
Depress Mouth Corners	Click	EP3	1	100.0	100.0	0.0
Tongue Out	Zoom In, Zoom Out	EP11, EP9, EP4	4	25.0	90.0	65.0

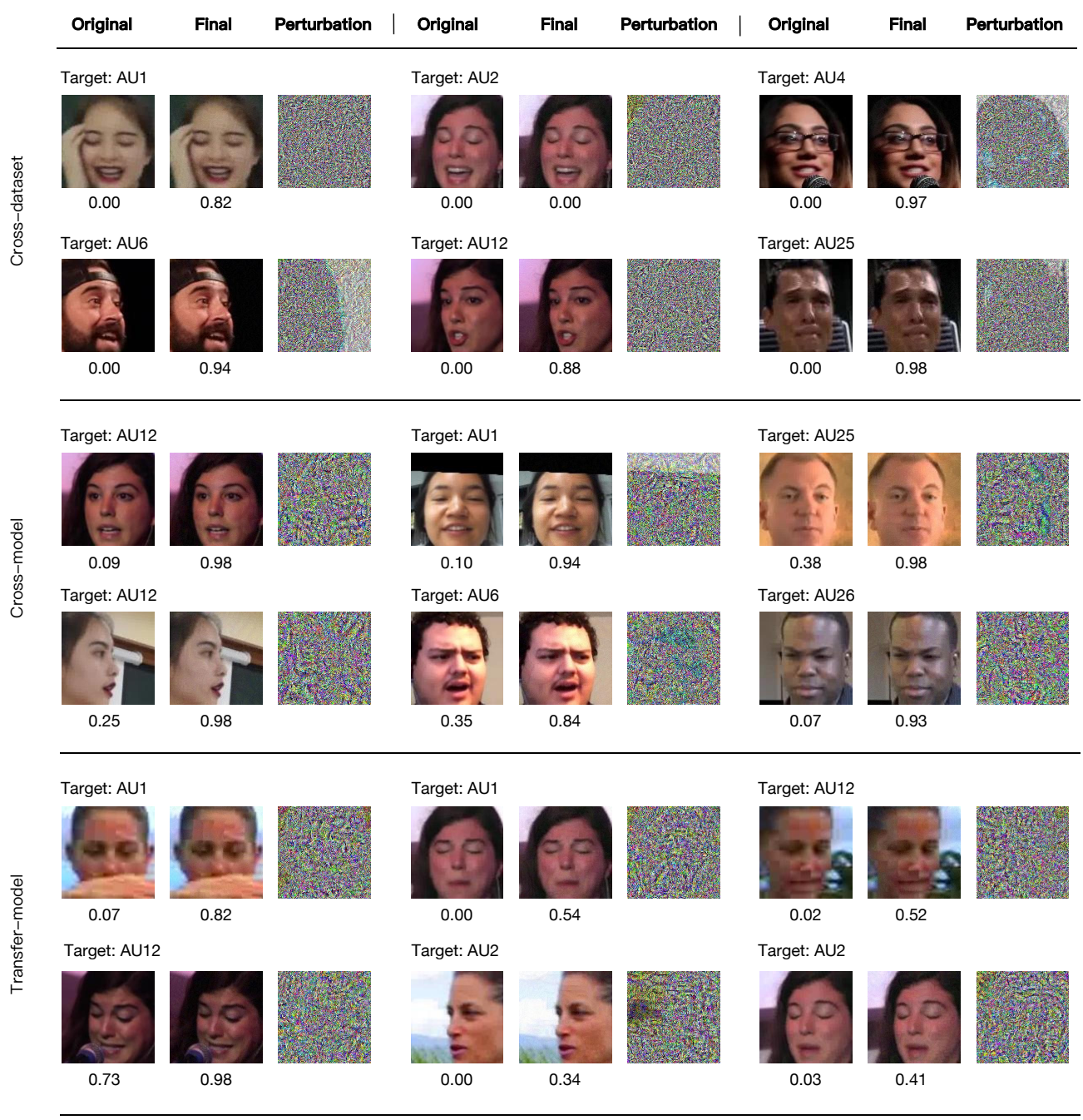

Figure 5: Additional qualitative examples from the three offline validation settings: cross-dataset, cross-model, and model transfer. Each triplet shows the original input, the assisted output, and a normalized perturbation visualization.