TRACE | UNIVERSITY OF CAMBRIDGE

WHITEPAPER • VERSION 1.3 • 4 NOVEMBER 2026

Designing for Hybrid Human–AI Workforces

A human-centred architecture for the division of labour between humans and agents

Organisations are adding agents to workflows that were designed for humans. The boundary between human and machine work is rarely designed. Tasks are often handed over whole, capabilities often go unmeasured, and the share of work humans keep is often set by the allocation rule rather than chosen. This report presents three deployment mechanisms for designing that boundary, so that the share humans keep is a decision.

1
Human API

Agents request knowledge, judgement and authorised decisions from humans. Each call records its source, its cost and the response.

2
Skill inference

Human and agent capabilities are estimated on one shared skill taxonomy. Graded work updates capability estimates, with uncertainty shown.

3
Dynamic controls

The right human share can differ by organisation, domain and task. The organisation sets a minimum share, and a dynamic control adjusts allocation and records its estimated output cost.

About this work

This paper comes from the TRACE Lab, University of Cambridge. The lab researches trustworthy AI systems for high-stakes environments and explores how agents can be effectively orchestrated alongside human workers. It operates within the Centre for Human-Inspired Artificial Intelligence (CHIA), whose aim is to advance AI for the benefit of humanity.

As organisations move from isolated AI tools to agents that act inside live workflows, whether those systems can be trusted becomes an operating requirement. The organisation must be able to say who did the work, why it was allocated to them, and who is accountable for the outcome. This paper aims to make those questions answerable in deployed systems.

Questions. Questions about this paper can be directed to the TRACE Lab, Centre for Human-Inspired AI, University of Cambridge: Kostadin Devedzhiev (kgd26@cam.ac.uk), Umang Bhatt (usb20@cam.ac.uk) and Ariella Shulman (as3732@cam.ac.uk).

Acknowledgements. This paper was developed in collaboration with Accenture Singapore, whose input informed its discussion of implementation in Singaporean organisations, and who supported the work financially.

Revision reference. Version 1.3, 4 November 2026. Revised claims and citations against their sources and the experiment data. Corrected the Healthcare skill count and clarified the trade-off rate. Reworded passages flagged as matching web text, and added a note on the use of AI tools. Version 1.2, 13 October 2026. Incorporated Accenture's detailed wording feedback: standardised human and skills-profile terminology; clarified the unit of work, typed-call delivery, escalation and routing assumptions; updated the Singapore agency reference to SWDA; and specified the experiments' taxonomy and the meaning of model beliefs. Version 1.1, 7 October 2026, introduced deployment-mechanism terminology, meaningful accountability, business implications, the practical pilot sequence and qualifications to simulated output-cost and capability-preservation claims.

Contents

About this work 2

1 Introduction 4
- 1.1 The problem 4
- 1.2 What this paper adds 5
- 1.3 Scope 6
- 1.4 The unit of work 6
- 1.5 How to read this paper 7

2 The Human API 8
- 2.1 The interface 8
- 2.2 Example: a care pathway 10
- 2.3 The typed call, and why free-form dialogue is out of scope 12
- 2.4 Discussion 13
- 2.5 Limitations 15

3 Skill inference 16
- 3.1 O*NET and Singapore Skills Frameworks 17
- 3.2 Skills profiles 18
- 3.3 Base algorithm: single-Gaussian population model 19
- 3.4 Worked example: credit memo evaluation 19
- 3.5 The mixture: a Gaussian mixture model over the population 20
- 3.6 Implementation and open-source package 21
- 3.7 Experiment 1: recovering skills profiles on O*NET 23
- 3.8 Experiment 2: orchestrating human–AI teams 26
- 3.9 Discussion 29
- 3.10 Limitations 30

4 Dynamic controls 32
- 4.1 The problem with a fixed human–AI share 32
- 4.2 Reasons for maintaining human participation 33
- 4.3 Setting the human share, then holding it 33
- 4.4 Worked example: the credit officer 34
- 4.5 The algorithm 35
- 4.6 Experiment 3: what the floor costs 36
- 4.7 Discussion 38
- 4.8 Limitations 39

5 Conclusion 40
- 5.1 What would have to be true, and how to find out 40
- 5.2 A practical pilot sequence 42
- 5.3 Closing 43

Bibliography 45

1 Introduction

1.1 The problem

Consider a bank's credit process. For every application a memorandum is written: the financials are analysed, the exceptions are explained, the risk is sized, and the decision is taken and signed by somebody who holds the authority to sign it. The process was designed around humans, and it says who writes, who reviews, who approves, and what each of them is answerable for. Suppose an agent now drafts the memos overnight. The analysts arrive to a stack of drafts and spend the day reviewing them. The work is faster. Three other aspects of the process have also changed without being explicitly designed. The analyst's job has become review rather than analysis, so the judgement that reviewing depends on is no longer being practised. Where the agent needed something only a human knew, such as the rationale behind a past exception, it either guessed or asked in a message that no system recorded. The share of work performed by humans is therefore determined by what the drafting tool does not automate.

This paper addresses that problem. Organisations are adding agents to workflows that were designed for humans, and the boundary between human and machine work is rarely designed. It tends to be set by tool adoption: by which tasks a given tool covers, and by the order in which the tools arrived. Three consequences follow. The organisation usually has no systematic record of any of them.

- **Work is handed over whole.** A task is either given to the agent or kept with a human. Where the agent needs information or authority that only a human holds, it obtains it informally, for example by phoning or messaging someone [6], or proceeds without it. Often nobody records who was asked, what they were asked for, or what the asking cost. When something goes wrong, there may be no record of who made the relevant decision.
- **Capability goes unmeasured.** Humans are described by job titles and CVs and agents by model cards, and the two are not directly comparable. Which actor should take a given piece of work is therefore often settled by assumption, and whether a human and an agent are complementary on that work, or whether one is better, is rarely measured. A meta-analysis of 106 experiments found that human–machine combinations performed worse on average than the better of the two alone, and gained where the human alone had outperformed the system [30]. Without measuring complementarity, an organisation cannot determine which of these cases applies.
- **The human share is not chosen.** Often nobody sets the share of work that stays with humans, so its cost is unknown. Skills that are no longer practised decay [4], and knowledge workers who trust generative AI report putting less critical-thinking effort into the tasks they hand to it [18]. The organisation may therefore lose capabilities that it needs when the agent is wrong or unavailable, without having explicitly chosen to do so.

This paper uses *trustworthy* in a specific sense. Here, trust means that reliance on the system is justified and that the basis for this reliance can be explained. It does not mean confidence in the system or its accuracy. Four conditions are required. The system does what is claimed of it. Somebody can see why it acted as it did. One named person answers for the outcome. A person can still refuse. When these conditions are applied to a model, responsibility for satisfying them lies with the model's developer. This paper asks them of the division of labour into which the model is deployed, because the three consequences above break them there. Accountability, explanation and the right to intervene are properties of the arrangement, not only of the model.

Meaningful accountability requires more than naming a human responsible for an outcome. That human must have the relevant skills and judgement, enough context and time to assess the work, and the authority to refuse, modify or reject it, or escalate it to someone suitably qualified. These conditions must be designed into the workflow before work is assigned.

1.2 What this paper adds

This paper presents three deployment mechanisms, one for each consequence. They are components an organisation can adopt.

Leaders introducing agents are redesigning how their organisation works. They need to decide which capabilities must remain with humans, where human judgement and authority are required, and how to evaluate the resulting division of labour. The three mechanisms make these decisions observable and reviewable: record the exchanges, estimate capability with its uncertainty, and manage human participation alongside current output. Their business value is the ability to pursue productivity while deliberately maintaining the capacity to scrutinise decisions, develop staff and respond when an agent fails.

- **The Human API** (Chapter 2) answers the first. It is a standard, bidirectional interface through which agents request data, decisions and validations from named humans, and humans assign work to agents, as structured calls. Its main effect is measurement: it records every exchange between humans and agents, which today often goes unrecorded, so that who was asked, for what, at what cost and with what consent is on file. Routing, traceability and accountability all depend on that record.
- **Skill inference** (Chapter 3) answers the second. It estimates the capabilities of humans and agents on one shared skill taxonomy, from the outcomes of routine work rather than from assessments, with an explicit measure of uncertainty that separates low proficiency from too little evidence. Comparable estimates let an allocation be defended and complementarity be measured.
- **Dynamic controls** (Chapter 4) answer the third. The organisation sets a minimum share of work that stays with humans as a group. The control holds that share as

work arrives and reports what holding it costs in output, so the share is a recorded decision with a known cost.

The three mechanisms rest on different kinds of evidence. The Human API is specified and illustrated in Chapter 2; its formal treatment and empirical study are the subject of a companion paper in preparation [2]. Skill inference and dynamic controls are evaluated in simulation on the O*NET taxonomy in Chapters 3 and 4. The data are synthetic, so those experiments show how the mechanisms behave under their assumptions, not numbers an organisation should expect. Chapter 5 names the assumptions the mechanisms rest on, the observation that would confirm or refute each, and what an organisation needs in place to run a pilot.

Together, the three components form an iterative process: Human API calls generate recorded task outcomes, skill inference uses these outcomes to update profiles and inform routing, and dynamic controls maintain the chosen human share. Each can be used alone, and an organisation can adopt them in any order. All three read and write one shared skill vocabulary. Singapore's national Skills Frameworks [28] are the target vocabulary for deployments in Singapore, and Chapter 3 works through the Skills Framework for Financial Services in particular.

1.3 Scope

The architecture is for *agentic* deployments: software that executes a workflow, or part of one, and can initiate requests of its own. In that case the division of labour has to be designed, because the agent must obtain information and authority from humans, and it needs an interface to do so. Where AI is used as a tool that a human invokes, the human already holds the work and initiates every exchange, and the Human API's agent-to-human direction has no role. Two of the three mechanisms carry over to that setting regardless. Skill inference scores any actor whose task outcomes are graded, whether the actor is a human, an agent or a tool-assisted human, and dynamic controls act on any allocation of tasks between humans and systems. The Human API, and the loop the three form, are specific to agentic deployment.

The paper also presumes an *established* process: one whose steps, decision points and information needs are known well enough to be written down, as a credit process or a clinical pathway is. Each mechanism says what it needs from that process, and the Human API in particular depends on it (Section 2.1). Discovering where the information gaps in an undocumented workflow sit is a different problem and is out of scope here.

1.4 The unit of work

This paper measures work in units. For a human, the unit is the piece of work they hold end to end and are accountable for, such as preparing a credit memo or conducting a patient interaction within a care pathway. The patient is the subject of care, not a unit of work. The human's unit includes the context, practice and ownership needed to exercise

judgement, and completing it provides practice relevant to the next case. For an agent, an assignable unit may be more granular: the part covered by one routed task or typed call, such as analysing one set of financials, retrieving an observation from a patient interaction, or requesting approval for one application. Introducing an agent makes the workflow more granular by separating a complete piece of work into assignable tasks. It is the workflow that is divided, not the patient. Information must pass between the human and the system at each task handoff.

The Human API of Chapter 2 operates at these handoffs. Skill inference in Chapter 3 estimates skills from graded task outcomes. The human share maintained in Chapter 4 is a share of those assignable tasks. The chosen level of granularity therefore determines what the measurements mean and which tasks can be assigned to an agent. Finer task definitions can report a different amount of work as automated or retained by humans even when the underlying process is unchanged, so comparisons require a consistent task definition. Granularity has a limit: the human accountable for the complete piece of work must retain enough context and substantive involvement to exercise judgement and practise the relevant skills. Isolated fragments may not provide either. The three mechanisms support an explicit decision about that limit; the humans accountable for the process must set it.

1.5 How to read this paper

The credit process runs through the paper as its working example: the Human API routes its calls in Chapter 2, skill inference scores its analysts in Chapter 3, and dynamic controls hold its human share in Chapter 4. A second, clinical example in Chapter 2 shows the Human API through patient interactions rather than document preparation, as the care pathway passes between care settings over months. Chapter 5 states what would have to be true for the three to work in practice and how an organisation would find out.

Business readers can start with Sections 1.1–1.2, the care pathway in Section 2.2, the worked control example in Section 4.4, and the implications for business leaders at the end of each core chapter. Section 5.2 sets out a practical pilot sequence. Readers assessing the technical basis can follow the full methods, algorithms and experiments.

2 The Human API

This chapter summarises the Human API protocol. The full protocol and its formal and empirical study are the subject of a companion paper in preparation [2].

For a process owner, the decision is where a workflow needs human knowledge or authority, who can supply it, and how their involvement will be recorded and reviewed.

2.1 The interface

The Human API provides a structured record of exchanges between humans and agents. This section describes that record and the rest of the chapter describes its uses. Today, an agent's request for human information or authority may arrive as a phone call or a chat message [6]. Those channels may retain text, but often do not capture consistent fields for request type, responder, cost, reliability, consent and downstream use. The Human API adds that structure; it does not prescribe a particular communication channel. A typed call can be presented through a familiar interface while the system records the exchange. Routing a request to the right human, tracing a decision to its inputs and evaluating accountability are all easier with the structured record.

Agents may require information or authority that is not available through existing databases or APIs [2]. This includes:

- **Physical world state:** Real-time physical observations (e.g., environmental conditions) that are not logged in digital systems or accessible via autonomous sensors.
- **Tacit institutional knowledge:** Contextual history and unwritten organisational practices, such as the rationale behind historical exceptions or other context-sensitive organisational knowledge.
- **Local norms and values:** Contextual judgement calls regarding organisational risk tolerance and policy appropriateness.
- **Formal authority:** Decisions that policy or regulation reserves for an authorised human. Supervisory guidance such as MAS FEAT expects the use of AI in decisions to be approved by an appropriate internal authority [22], and a delegation of authority can specify which role, committee or individual may approve a decision.

Physical observations and institutional knowledge may become more digitised over time. This design reserves value judgements and formal approvals for humans.

Two conditions for use. The protocol presumes the established process of Section 1.3: it must be possible to say in advance which step needs an observation from a human, which needs the history of an exception, and which needs a signature. A credit memo and a clinical pathway both meet this condition, which makes a registry of calls possible. The protocol cannot be applied to a workflow whose information needs are unknown. The

second condition concerns how information and skills are distinguished. A call requests information or a decision from a human. It does not measure capability. Capability is estimated in Chapter 3 from how the outcomes of calls are graded. A nurse who supplies a reading is not assessed on it. Profiles are built only from graded outcomes.

Typed calls. The calls an agent may make are defined in advance in a *registry*, which lists each kind of request, the sources it may go to, and what it returns. The Human API routes each request as a **typed call** drawn from that registry. Here, *typed* means that the request has a predefined category and response format, not that the responder must enter computer code. The delivery channel is an implementation choice: a chat message with a structured reply, an app notification, a web form or a staff task screen can all carry a typed call. For example, the patient's daughter might receive an agreed app or text-message prompt asking whether a prescription was collected, with *yes*, *no*, *not sure* and *decline* response options. The system would store her reply alongside the request's structured fields. This is an illustrative interface, not an implemented feature evaluated here. A call records six attributes. Each attribute supports a specific auditing or allocation decision.

- **Call type**, so that what was asked for is explicit and repeatable: a reading, a history, a judgement, an approval, each with a defined return.
- **Source identifier**, so that the record names which human was asked, and workload, consent and contribution can be attributed to them.
- **Cost**, in the human's time, so that the time taken is counted against the value of the answer, and the agent's choice of whom to ask can be judged.
- **Reliability estimate**, so that an answer is weighted by how good this source has been at this kind of question, and the weighting can be checked.
- **Consent status**, so that acceptance, modification and refusal are each on the record, and a refusal is recorded as a refusal rather than as a missing answer.
- **Audit record**, so that how the answer was used can be traced.

The audit record holds the call as issued, the reply as given, the source, the time, the cost charged, and whether the request was accepted, modified or declined. It also links to the downstream task the reply fed and to what the agent did with it. A reviewer can reconstruct from one record which humans were asked what, what they answered, and how the answer was used. A complete record helps a regulator reconstruct how the process was executed. A deployment should preserve original calls and replies and append any corrections with their author and time, linked to the downstream decisions they affect. This is an implementation requirement for a traceable audit record, rather than an empirical property established here. The protocol also works in reverse, allowing human workers to assign tasks to agents through the same interface. Figure 1 shows the two directions.

Auditability and operational metrics

Because every exchange routed through the protocol is a typed call, it can be counted. Unlike text transcripts, the record yields operational metrics directly:

- **Workload:** Tracks request volume per individual or team to help prevent over-allocation.
- **Consent and refusal:** Records request modifications and rejections, which can reveal boundary constraints and selection bias.
- **Contribution tracking:** Attributes specific human inputs to downstream task outcomes for post-hoc analysis.
- **Profile freshness:** Links each call to the task outcome it fed. Where a responder's contribution is graded on named skills, the grade can update their profile (Chapter 3), and the controls of Chapter 4 act on those estimates.

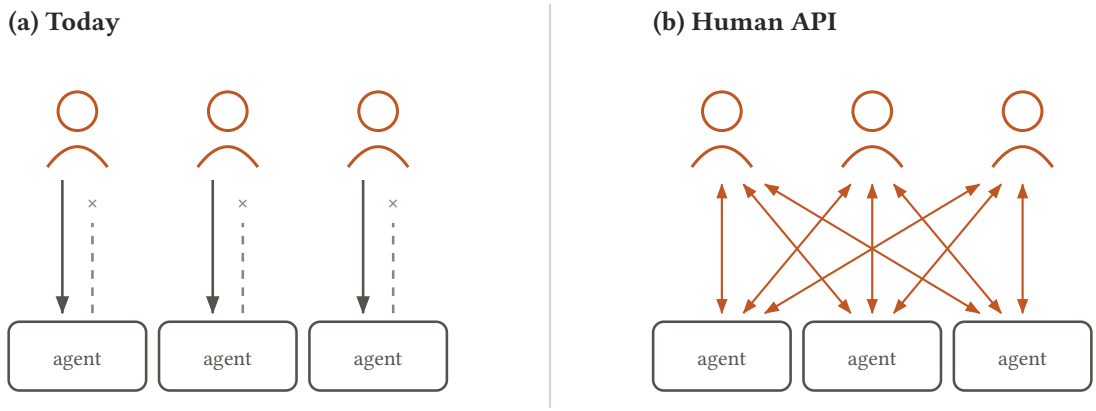


Figure 1: The direction of calls. (a) Humans drive agents. (b) Calls flow both ways. Humans are called for inputs; they need not route work by hand.

2.2 Example: a care pathway

Consider a patient with heart failure, followed from admission to the sixth month after discharge. When she arrives, the emergency department is full. The clinician who sees her needs a usable summary of her record. Over the months that follow, her care passes through a ward team, a community care centre and her own home. Each handoff creates a risk that relevant information will not be transferred. What was decided on the ward is not always known at the community care centre, and events at home between visits may not reach the care team. A missed refill or a rising weight can go unnoticed until her next appointment. Clinical judgement is available at each site, but no single participant follows the case across the entire pathway.

The Human API adds an agent that follows the case throughout, with a narrow job. The care pathway agent watches the record and retrieves and prepares what the next human will need. It tracks open items, asks a human for judgements and observations that only a human can give, and records decisions. It makes no clinical decisions itself. A human remains responsible for the case throughout the pathway. In this scenario that is the consultant on the ward and, after discharge, the community nurse. They can review the whole record and make the clinical decisions within their authority. The agent gives them a synthesis of the record and a list of open items. Two gains are expected: less clinician time spent reassembling the case at each handoff, and earlier notice of events between visits. Neither is demonstrated here. A pilot of the kind Chapter 5 describes would measure both.

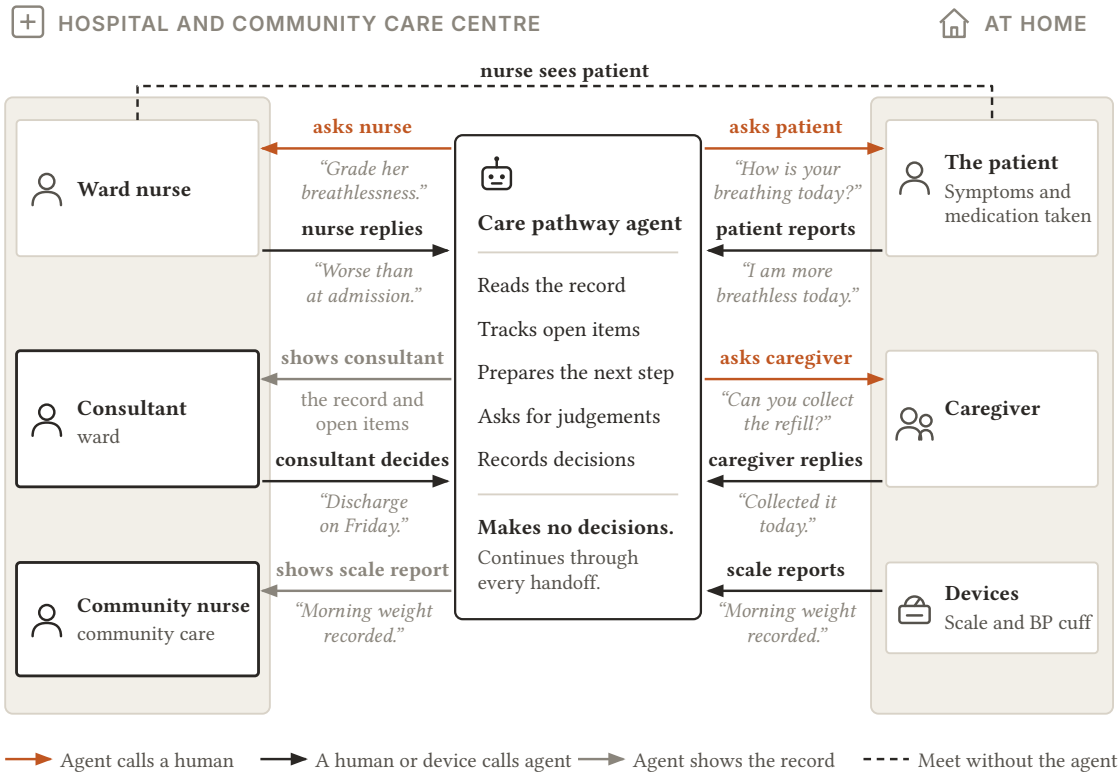


Figure 2: The Human API in a hospital care pathway. Every call is recorded, including who was asked and their response. The consultant, and later the community nurse, can view the full record, and anyone called can decline a request. The agent does not make clinical decisions, and it does not stand between the nurse and the patient. The Human API provides a structured way to involve humans in care workflows and supports human oversight.

Figure 2 shows the arrangement. The agent calls the ward nurse, the patient and her daughter for information and decisions. The nurse, the patient, the daughter and the scale report to the agent. The agent shows the consultant the record and the open items, and the consultant takes the clinical decisions. After discharge the community nurse owns the case, and the agent shows the community nurse what the scale reported. The ward nurse sees the patient in person; the agent only asks the nurse what they found.

Each call records its source and its cost.

Context included with each call. Each call includes the context required by the responder to answer it appropriately: who the patient is, where she is in the pathway, what the question is for, and what has already been established that bears on it. The nurse asked to grade breathlessness sees that this is the heart-failure patient admitted on Tuesday, that the question is being asked because the weight reported from home has risen, and what the last grading was. A responder who needs more can ask for it, and the asking is itself a call on the record. A responder does not receive the whole case. The aim is to request human input only for the part of the task that requires human involvement.

When the whole case requires human review. Requests for specific inputs are the default, with exceptions. The whole case goes to the accountable human for review when:

- the responses coming back disagree with each other;
- a reading crosses a threshold the pathway defines as requiring review;
- the patient or caregiver asks for a human;
- a responder cannot give a sufficiently reliable answer to the specific question without reviewing more of the case.

The final condition concerns whether the supplied context is sufficient for a reliable answer. A responder may, for example, be unable to interpret a reported change in breathlessness without the recent clinical history. The responder can request more context, decline the specific request or escalate it for a review of the complete case. Refusal and escalation should be recorded separately so that declining an input is not confused with making a clinical decision. The accountable human then decides how to proceed. The consultant on the ward, and the community nurse after discharge, can review the complete case at any time; the agent may request this review but cannot prevent it.

2.3 The typed call, and why free-form dialogue is out of scope

Figure 3 shows one request in both forms: as a typed call drawn from the registry, and pursued in free-form dialogue. The protocol uses the typed call, and free-form dialogue is out of scope here. An agent could be allowed to ask humans anything in natural language, and a companion study in preparation examines both forms [2]. A free-form exchange does not provide the protocol's structured fields by default. Free-form dialogue allows follow-up questions the registry did not anticipate, but measuring its cost, routing, consent and outcome requires those fields to be captured explicitly. The typed call supplies that structure. A conversational front end could also collect the required metadata; unrestricted dialogue without it is outside the protocol described here.

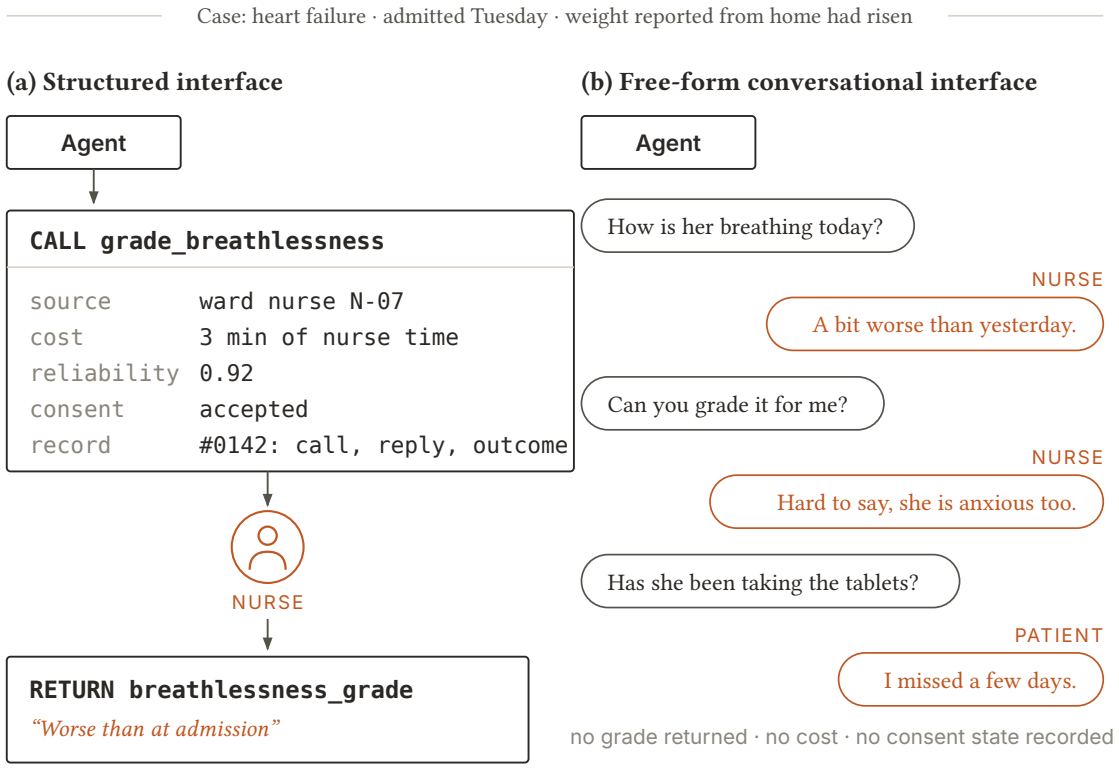


Figure 3: The two forms of a request. (a) A typed call, with its six attributes. (b) The same request pursued in free-form dialogue. The typed call supplies the fields needed for costing, routing and logging; the dialogue shown does not.

The call is the first one in Figure 2. Form adapted from [2]; dialogue, returned value, reliability and record values are illustrative.

Takeaways

- The Human API records the exchanges routed through it. That record supports routing, traceability and accountability.
- Some information and authority exist only in humans. A typed call requests that knowledge or an authorised decision. It does not measure capability.
- Every call records type, source, cost, reliability, consent and an audit record linking the response to the decisions it fed.
- Workers can decline or modify requests, and those actions stay on the record. The interface records refusals and modifications, which supports oversight.

2.4 Discussion

Credit memo example. In the credit memo of Chapter 1, an agent drafts the memo overnight. Where the draft depends on the rationale behind a past exception, the agent sends a typed call to the analyst who handled it. The analyst answers, modifies the

request, or declines. Where the memo needs approval, the agent sends a call to the credit officer who holds sign-off authority. Each call records who was asked, what it cost, whether they consented, and what they returned. The analyst reviews the parts of the memo that need a human rather than the whole draft, and can ask for the whole memo when that is needed to answer for it. Grades of the analyst's contributions linked to those calls supply evidence for the skills profiles of Chapter 3.

From full-task deferral to targeted human input. Reject-option and learning-to-defer methods use binary handoffs: each case is either given to a human or handled by the model [8, 23]. Later work extends deferral to several experts and to caps on each expert's workload, but the unit of deferral is still the whole task [1, 31]. The Human API instead allows agents to request specific inputs from selected individuals while continuing to execute the task [2].

Accountability and governance. By recording the source, authority level, and context for every request, typed calls make decision ownership traceable. Meaningful accountability also requires the skills, judgement, context and intervention authority defined in Chapter 1; a recorded name or signature alone does not establish it. This aligns with supervisory guidance, such as the Monetary Authority of Singapore (MAS) FEAT principles, which expect firms to remain accountable for automated decision systems and to place their use under an appropriate internal approving authority [22].

The same mechanism can represent an organisation's **delegation of authority**. For example, a bank's delegation of authority might say who may approve what by the size of the exposure and its sector, with the largest exposures going to a committee. In the Human API an approval is a typed call routed under that policy. Its record shows who approved, under which delegation, when, and on what information. No attributes beyond the six are needed: the policy sets the source, and the audit record holds the rest.

Accountability also requires that humans can decline or modify requests, with those actions kept in the audit log, and that the audit trail is open to the relevant workers, accountable officers and authorised regulators, subject to confidentiality.

Implication for business leaders. Managers can use the Human API to see whose knowledge and authority a workflow depends on, how much time requests consume, and where refusals or escalation reveal poorly designed handoffs. The business case is to reduce coordination effort and rework while preserving decision quality and humans' ability to intervene. These benefits require testing: run comparable cases through typed calls and existing channels, track decision quality, time to decision and rework independently, and link outcomes to the recorded calls. A complete record enables evaluation; it does not itself improve decisions. Chapter 5 sets out the pilot.

2.5 Limitations

The protocol is specified and illustrated in this chapter rather than measured in it, and three limits follow. First, the registry restricts the predefined requests available to an agent. A request not anticipated by the registry needs a new call definition or escalation outside the predefined set. Second, the cost and reliability estimates are supplied to the protocol; the protocol does not establish that those estimates are accurate. If a responder's time cost is underestimated or their reliability is overestimated, the routing rule may favour that responder too often. Conversely, an overestimated cost or underestimated reliability may cause a suitable responder to be overlooked. These inputs must be calibrated against observed response times and independently assessed answer quality. Third, the human factors that determine whether the interface works in practice are not evaluated here. These include cognitive fatigue, the proportion of requests humans decline, and whether an organisation treats a refusal as a legitimate exercise of consent or as a negative performance signal. The refusal proportion means declined requests divided by requests received over a stated period; it can be examined by request type and responder alongside reasons for refusal, workload and escalation. A high proportion alone does not show poor performance. An empirical study of the protocol is being prepared in the companion paper [2]. Its behaviour in a live process is untested, as is the outcome claim of Section 2.4. This chapter shows that exchanges can be recorded; it does not show that recording them improves decisions.

3 Skill inference

This chapter summarises the skill inference method of a companion paper that is currently in preparation [11]. Sections 3.3–3.6 describe the method and Section 3.9 its implications. The estimator's own evaluation is in that paper. The two experiments in Sections 3.7–3.8 were run for this paper on the population it uses.

For a manager, the decision is who should receive a task, where further evidence is needed, and which development opportunities the allocation should create.

Chapter 2 defines the interface for human–agent calls. Effective orchestration also requires knowing *which* actor should receive each request. Static job descriptions, CVs, and fixed AI model cards are incomparable while frequent formal benchmarking is expensive and can become outdated quickly.

Skills profiles can instead be inferred from routine work. Each completed task provides observations for a subset of skills. Known correlations between skills allow one task outcome to update related skills. Both are assumptions about real work rather than findings of this chapter; Section 3.10 names them with the others the method rests on, and Chapter 5 says how each would be tested. Tracking variance as well as mean capability also helps distinguish low proficiency from high uncertainty caused by limited observations. Humans and agents can then be evaluated using the same skill taxonomy, and tasks can be routed using their updated performance estimates. Figure 4 shows this process.

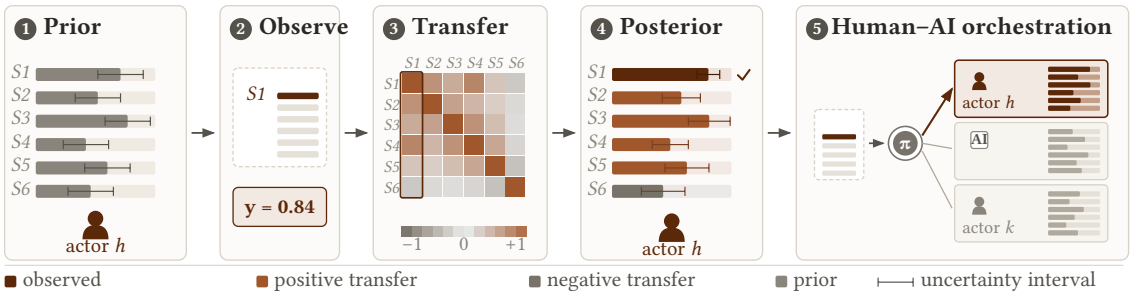


Figure 4: Skill inference and orchestration pipeline. Stages **1–4** infer skills profiles from observed performance. Stage **5** uses these profiles to route tasks. Actors start at baseline population averages, and subsequent task outcomes refine skill estimates while reducing uncertainty across correlated dimensions. Profiles are learned from routine work and then used to route it.

The inference approach is validated through two simulated experiments using the O*NET taxonomy. Experiment 1 (Section 3.7) evaluates profile recovery from noisy observations, while Experiment 2 (Section 3.8) evaluates task routing efficiency under cold-start conditions across mixed human–agent populations.

3.1 O*NET and Singapore Skills Frameworks

This paper considers three skill taxonomies. O*NET supplies the data for its simulation experiments; the two Singapore Skills Frameworks provide vocabularies for deployment.[1]

O*NET: The U.S. Department of Labor's occupational database (Release 30.2) rates 894 occupations on all 120 standardised skills, knowledge areas, and abilities [24]. Its dense feature matrix provides ground-truth data for empirical validation and baseline synthetic experiments. It is also the basis of widely cited estimates of which occupations are exposed to language models [12].

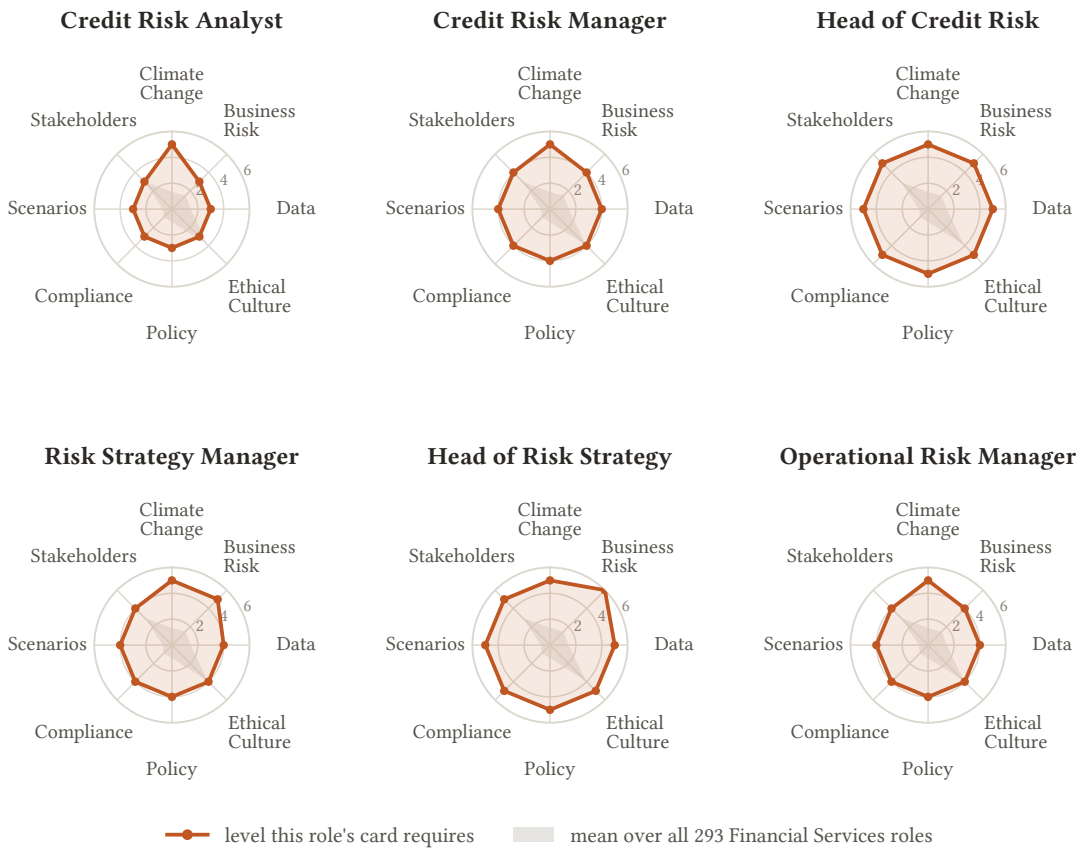


Figure 5: Skills profiles across financial roles. Profiles evaluate competency across eight sample skill dimensions defined in the Skills Framework for Financial Services [15]. Shaded regions reflect population averages across 293 industry roles. One shared taxonomy scores every role, so profiles are directly comparable.

Skills Framework for Financial Services (Singapore): Developed jointly by the Institute of Banking and Finance, MAS and Singapore's national skills and workforce agencies, now combined in the Skills and Workforce Development Agency (SWDA), this taxonomy details 293 job roles against 140 sector-specific skills in the Skills Framework

[1] SWDA was established on 1 July 2026 through the merger of SkillsFuture Singapore and Workforce Singapore; see Ministry of Manpower, 24 June 2026.

dataset used in this paper [15]. Each role profile lists the skills the role requires and a target proficiency level for each, on a scale of one to six. It is the target vocabulary for deployments in Singapore, and Figure 5 shows six role profiles expressed in it.

Skills Framework for Healthcare (Singapore): The same national programme publishes a Healthcare framework, covering 77 job roles against 137 sector-specific skills in the Skills Framework dataset used in this paper [28]. It is the vocabulary in which the clinical example of Chapter 2 would be written for a Singapore deployment. Financial Services and Healthcare are used throughout for the same reason, that they are the two sectors where the division of labour in this paper is easiest to recognise, and the estimator behaves the same way in both.

Which taxonomy fits. The three serve different purposes and none of them is assumed to be the right instrument for a given organisation. O*NET is used for the three simulation experiments reported in this paper (Sections 3.7, 3.8 and 4.6) because it is dense: every occupation is rated on every one of the 120 features, which is what a simulation needs to draw realistic populations and what the transfer model needs to fit a correlation structure. It was built for career guidance in the US labour market, and it decomposes roles into skills, knowledge and abilities that are free of organisational context. This makes it applicable across 894 occupations but less directly applicable to organisational decision-making. Managers typically do not assign or train abstract capabilities such as "deductive reasoning" directly. Organisations instead commonly use competency frameworks, which name contextual capabilities ("structures a credit proposal", "assesses collateral") and map more naturally onto roles, adjacency between roles, and development paths. The Skills Frameworks provide an intermediate level of abstraction, because they are both sector-specific and role-indexed, which is why they are the deployment vocabulary here. The estimator works with any taxonomy that provides named dimensions, graded outcomes on some of them, and a population for fitting the correlations. What changes with the taxonomy is the correlation prior. A competency framework has no analyst ratings to fit it from, so the population structure would have to be learned from the organisation's own graded outcomes, which means a longer cold start and a prior that is at first diagonal. The method is unchanged but starts with less information.

3.2 Skills profiles

Every actor a (human or agent) is represented by a skills profile over a taxonomy of K skills. The profile consists of a mean competency vector $\boldsymbol{\mu}_a \in [0,1]^K$ and a covariance matrix $\boldsymbol{\Sigma}_a$. The diagonal entries of $\boldsymbol{\Sigma}_a$ represent variance (uncertainty), while the off-diagonal entries model pairwise skill correlations. Figure 5 illustrates sample skills profiles mapped across credit approval roles within the Financial Services framework.

Representing an actor by a mean and a variance, and updating both after each observed outcome, follows Bayesian rating systems such as TrueSkill [13]. Knowledge tracing in education maintains a comparable belief over mastery that is updated after each observed outcome [9]. Those methods track one score per player or one skill at a time. The profile

here tracks all K skills jointly, and the covariance structure allows evidence from one observed skill to update correlated skills in the profile.

3.3 Base algorithm: single-Gaussian population model

The base estimator maintains a unified covariance structure calculated across the population. Actor skills profiles are tracked as Gaussian distributions, $\mathcal{N}(\boldsymbol{\mu}_a, \boldsymbol{\Sigma}_a)$, and updated sequentially using a Kalman filter framework [16].

When an actor completes a task associated with skill j producing outcome y_j, the profile updates as follows:

$$\mathbf{K} = \frac{\boldsymbol{\Sigma}_{:,j}}{\Sigma_{jj} + \sigma^2_{\text{obs}}}, \tag{3.1}$$

$$\boldsymbol{\mu} \leftarrow \boldsymbol{\mu} + \mathbf{K}(y_j - \mu_j), \tag{3.2}$$

$$\boldsymbol{\Sigma} \leftarrow \boldsymbol{\Sigma} - \mathbf{K}\boldsymbol{\Sigma}_{j,:}. \tag{3.3}$$

Equation (3.1) computes the Kalman gain based on current estimation variance and observation noise σ^2_{obs}. Equation (3.2) updates mean skill estimates proportional to the prediction error $(y_j - \mu_j)$, transferring updates to correlated skills via $\mathbf{K}$. Equation (3.3) reduces variance to reflect increased certainty. Algorithm 1 details the sequential update procedure.

Algorithm 1 Sequential single-Gaussian Kalman updates

Require: Population matrix $\mathbf{R}$, entity a, tasks $t_1, \ldots, t_k$ with active skills $\mathcal{S}_{t_i}$, observation noise σ_{obs}
Ensure: Skill vector $\boldsymbol{\mu}_a$, skill covariance $\boldsymbol{\Sigma}_a$ (unique per entity)

$\bar{\mathbf{r}} \leftarrow$ column means of $\mathbf{R}$ ▷ Population mean
$\boldsymbol{\Sigma} \leftarrow \frac{1}{N-1}(\mathbf{R} - \bar{\mathbf{R}})^\top(\mathbf{R} - \bar{\mathbf{R}})$ ▷ Population covariance
$\boldsymbol{\mu}_a \leftarrow \bar{\mathbf{r}}$, $\boldsymbol{\Sigma}_a \leftarrow \boldsymbol{\Sigma}$ ▷ Initialisation at population mean
for $i = 1, \ldots, k$ **do**
    Observe entity a on task t_i. Collect per-skill observations $\{y_j\}_{j \in \mathcal{S}_{t_i}}$
    for $j \in \mathcal{S}_{t_i}$ **do**
        $\mathbf{K} \leftarrow \boldsymbol{\Sigma}_a[:, j] \,/\, (\Sigma_{a,jj} + \sigma^2_{\text{obs}})$ ▷ Kalman gain, Eq. (3.1)
        $\boldsymbol{\mu}_a \leftarrow \boldsymbol{\mu}_a + \mathbf{K}(y_j - \mu_{a,j})$ ▷ Mean update, Eq. (3.2)
        $\boldsymbol{\Sigma}_a \leftarrow \boldsymbol{\Sigma}_a - \mathbf{K}\,\boldsymbol{\Sigma}_a[j, :]$ ▷ Covariance update, Eq. (3.3)
    end for
end for
return $\boldsymbol{\mu}_a$, $\boldsymbol{\Sigma}_a$

3.4 Worked example: credit memo evaluation

Consider a newly onboarded analyst drafting a credit memorandum. With no prior performance history, the analyst's profile initialises at the population baseline with broad uncertainty bounds. Upon review, a supervisor evaluates the memo's financial analysis component, assigning a score of 0.92 to the *economics and accounting* skill.

Using the update equations, this single observation modifies the analyst's broader profile, as Figure 6 shows: the mean estimate for *economics and accounting* adjusts toward the observed performance, while positively correlated dimensions (e.g., *administration and management*) adjust upward accordingly. Conversely, uncorrelated dimensions remain largely unchanged, and overall uncertainty across related skills decreases.

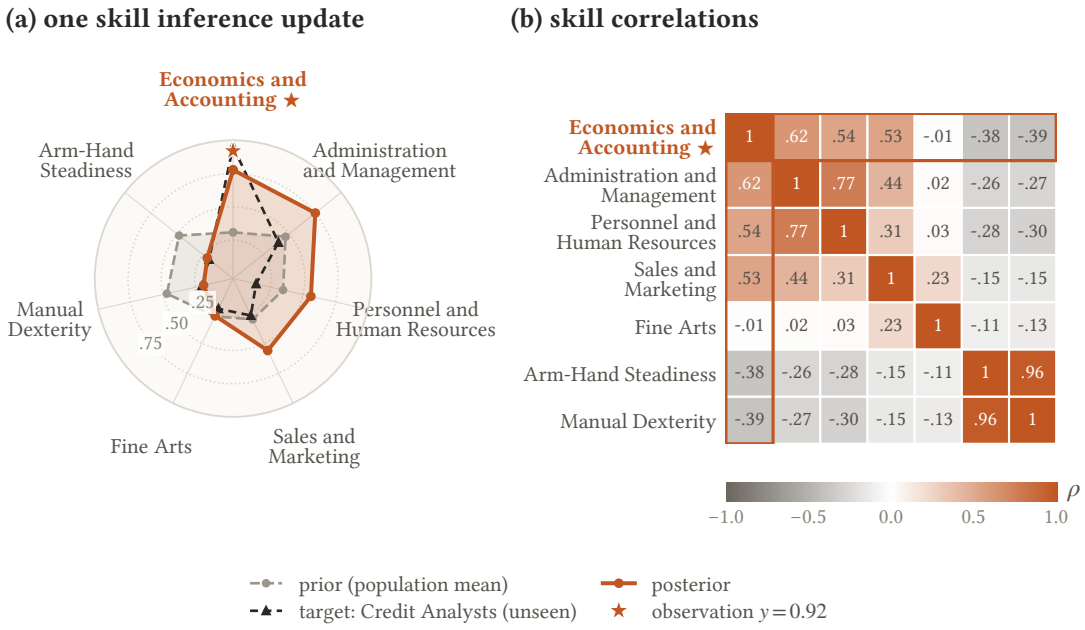


Figure 6: One observation moves the whole profile. (a) A single update, Equations (3.1)–(3.3), shown on seven of the 120 O*NET features. (b) The correlations among those seven that carry it. Correlations allow information from one graded task to update estimates for related skills.

O*NET 30.2, DOL/ETA, CC BY 4.0 [24].

3.5 The mixture: a Gaussian mixture model over the population

The quality of skill transfer depends on the correlations represented in the population model. A single covariance matrix assumes that the same skill relationships apply across the entire population. A population containing both humans and agents may have different skill correlation structures across actor types. For a human, reading and writing proficiency tend to rise together, whereas for a retrieval model, strong recall need not imply skill at tool use. Using a single covariance matrix for both groups may therefore represent neither population well, causing transfer to move estimates in the wrong direction. AI systems also differ among themselves: language models vary substantially in overall ability [14] and in distinct groups of skills [7].

To address this, the base estimator is extended to a **mixture model**. The extension changes only the population prior. Instead of fitting a single covariance to the whole population, the mixture fits M clusters by expectation–maximisation [10], each with its

own correlation structure. Each actor is then evaluated under all clusters simultaneously. All other components, including the Kalman update, remain unchanged. Rather than being assigned to a single cluster, a new actor maintains a belief state under every cluster together with a weight reflecting how likely that cluster is. Each observation serves two purposes: it updates every cluster's belief through the standard Kalman step, and it re-weights the clusters according to how well each one predicted the outcome. The model therefore estimates actor type jointly with skill levels, and skill transfer is based primarily on the correlation structure of the actor's inferred population. The allocation policy still reads a single level and confidence interval per skill, collapsed from the M beliefs. This interval additionally reflects disagreement between clusters: it remains wide while the actor's type is unknown and tightens as the weights concentrate. Algorithm 2 provides the pseudocode. It requires $\mathcal{O}(MK^2)$ operations per observation, and at $M = 1$ it reduces to Algorithm 1. The mixture size M can be selected empirically or via the Bayesian information criterion [27].

Algorithm 2 Mixture-of-Gaussians skill inference

Require: Population matrix $\mathbf{R}$, mixture size M, entity a, tasks $t_1, \ldots, t_k$ with active skills $\mathcal{S}_{t_i}$, and observation noise σ_{obs}
Ensure: Predicted skill vector $\boldsymbol{\mu}_a$, skill covariance $\boldsymbol{\Sigma}_a$

Fit GMM on $\mathbf{R}$ via EM: $\{\boldsymbol{\mu}_m, \boldsymbol{\Sigma}_m, \pi_m\}_{m=1}^{M}$
Initialise entity: $\boldsymbol{\mu}_{a,m} \leftarrow \boldsymbol{\mu}_m$, $\boldsymbol{\Sigma}_{a,m} \leftarrow \boldsymbol{\Sigma}_m$, $\pi_{a,m} \leftarrow \pi_m$ for $m = 1, \ldots, M$
for $i = 1, \ldots, k$ **do**
    Observe entity on task t_i and collect skill ratings $\{y_j\}_{j \in \mathcal{S}_{t_i}}$
    for $j \in \mathcal{S}_{t_i}$ **do**
        for $m = 1, \ldots, M$ **do**
            $\pi_{a,m} \leftarrow \pi_{a,m} \cdot \mathcal{N}\big(y_j \mid \mu_{a,m,j},\, \Sigma_{a,m,jj} + \sigma^2_{\text{obs}}\big)$ ▷ Mass update
            $\mathbf{K}_{a,m} \leftarrow \boldsymbol{\Sigma}_{a,m}[:, j] \,/\, (\Sigma_{a,m,jj} + \sigma^2_{\text{obs}})$ ▷ Kalman gain
            $\boldsymbol{\mu}_{a,m} \leftarrow \boldsymbol{\mu}_{a,m} + \mathbf{K}_{a,m}(y_j - \mu_{a,m,j})$ ▷ Skill transfer update
            $\boldsymbol{\Sigma}_{a,m} \leftarrow \boldsymbol{\Sigma}_{a,m} - \mathbf{K}_{a,m}\, \boldsymbol{\Sigma}_{a,m}[j, :]$ ▷ Uncertainty reduction
        end for
        Normalise: $\pi_{a,m} \leftarrow \pi_{a,m} / \sum_{m'=1}^{M} \pi_{a,m'}$ for $m = 1, \ldots, M$
    end for
end for
$\boldsymbol{\mu}_a \leftarrow \sum_m \pi_{a,m} \boldsymbol{\mu}_{a,m}$ ▷ Skill vector
$\boldsymbol{\Sigma}_a \leftarrow \sum_m \pi_{a,m} \left[\boldsymbol{\Sigma}_{a,m} + (\boldsymbol{\mu}_{a,m} - \boldsymbol{\mu}_a)(\boldsymbol{\mu}_{a,m} - \boldsymbol{\mu}_a)^\top\right]$ ▷ Skill covariance
return $\boldsymbol{\mu}_a$, $\boldsymbol{\Sigma}_a$

3.6 Implementation and open-source package

The toolkit is available as an open-source Python package, MIT licensed and installable via `pip`.[2] The package includes both skill inference algorithms, a held-out evaluation routine and plotting utilities. O*NET, ESCO, PIAAC and the Skills Framework are preloaded as populations, allowing a first profile to be constructed without user-supplied data. The documentation includes tutorials on O*NET, ESCO, PIAAC, LLM benchmarks and agent orchestration.[3]

[2] https://pypi.org/project/skillinfer/
[3] https://kostadindev.github.io/skillinfer/

The library is built around two main objects. A `Population` holds the fitted structure: it takes a table of entities by skills and estimates the population covariance of Algorithm 1 with Ledoit–Wolf shrinkage [17], a regularised form of the sample covariance. A `Profile` represents one actor and exposes two methods: `observe` updates the estimate with a single skill observation, and `predict` returns every skill with a confidence interval. The choice of estimator corresponds to a choice of prior and is passed as a single argument on the profile call: the single-Gaussian update by default, the mixture of Gaussians under `method="gmm-kalman"`, or one of the restricted priors compared in the next section.

Each `observe` call is a single $\mathcal{O}(K^2)$ rank-one update, completing in milliseconds on an ordinary CPU. Two observations produce estimates for the full 120-skills profile, with 95% confidence intervals for each skill. The observed skills have estimates close to their measured values. The unobserved skills keep wider intervals, and the narrowest of these belong to the skills the population correlates most strongly with what was observed. The following example shows the analyst from Section 3.4, scored on the memo's economics and writing components. It is here to show the shape of the interface a developer works with, which is two calls, `observe` and `predict`, rather than to make a claim about how much can be recovered from two observations; that is what Sections 3.7 and 3.8 measure.

```
import skillinfer

pop   = skillinfer.datasets.onet()   # 894 occupations x 120 skills
actor = pop.profile()                # new actor, nothing observed yet

actor.observe("Knowledge:Economics and Accounting", 0.92)
actor.observe("Skill:Writing", 0.85)
actor.predict()                      # every skill, with an interval
```

The output below shows five of the 120 returned rows, with the two observed skills marked:

	feature	mean	sd	95% interval
•	Knowledge:Economics and Accounting	0.92	0.01	[0.90, 0.94]
•	Skill:Writing	0.85	0.01	[0.83, 0.87]
	Knowledge:Administration and Management	0.86	0.14	[0.59, 1.00]
	Ability:Manual Dexterity	0.00	0.20	[0.00, 0.38]
	Knowledge:Fine Arts	0.28	0.12	[0.04, 0.52]

• watched

The two observed skills are set close to their measured values. Administration and management rises from an initial 0.46 to 0.86 because the population correlates it positively with both observations, while manual dexterity falls from 0.47 to the scale minimum because the population correlates it negatively with them. Fine arts barely moves from 0.27, confirming that the correlations drive the updates rather than a uniform shift, and every interval narrows. Manual dexterity is at the minimum of the scale and administration's interval extends to the upper bound. This is the bounded-scale limitation noted in Section 3.10: a Gaussian on a $[0, 1]$ scale is least accurate near the top of the range, where sign-off thresholds are set.

3.7 Experiment 1: recovering skills profiles on O*NET

This section and the next report two experiments. The first tests whether the estimator can recover an unseen profile from a small number of observations. The second tests whether the resulting profiles are accurate enough to route work between humans and machines.

Both experiments are simulations, as is the third in Chapter 4. The population data is real: the skill ratings, and the correlations fitted from them, come from O*NET's analyst files. All acting entities are generated: each actor's individual capability, the agents' advantages, the arriving tasks, the graded scores they produce, and whether a task was cleared.

A routing rule that withholds work when evidence is thin requires two properties from a profile: an estimate of what an actor can do, and an uncertainty estimate that indicates when the evidence is limited. This experiment evaluates whether a small number of noisy observations provides both.

Setup

Correlations are fitted on a random 80% of occupations, and all results are scored on the held-out occupations. The recovery target is an individual rather than an occupation: an actor is defined as their occupation's requirements plus a personal offset, $\boldsymbol{\theta} = \mathbf{x}_o + \boldsymbol{\varepsilon}$, $\varepsilon \sim \mathcal{N}(0, \sigma^2_{\text{ind}}\mathbf{P})$, and each observation adds noise, $y_j = \theta_j + \eta$, $\eta \sim \mathcal{N}(0, \sigma^2_{\text{obs}})$. Both spreads are set to 0.10 in this experiment, within the 0.15 median spread of incumbent ratings within an occupation in O*NET's analyst files, and the observation noise is swept in the analysis below. Each estimator starts from the occupation covariance plus the offset term, $\boldsymbol{\Sigma} + \sigma^2_{\text{ind}}\mathbf{P}$, with σ_{ind} taken as known. The offset follows the population's own correlation pattern $\mathbf{P}$, since individuals deviate from their occupational profile in correlated ways; an independent offset is reported as a lower bound. Each estimator receives a budget of 50 observations and is scored on the full 120-skills profile of every held-out actor a, observed skills included. Results are averaged over 10 trials of 40 actors each.

Metrics and baselines

Three metrics are computed, all over the full profile:

- **Recovery** $R(k)$, the share of starting squared error removed after k observations, pooled over actors rather than averaged per actor.

$$R(k) = 1 - \frac{\sum_a \|\boldsymbol{\mu}_a^{(k)} - \boldsymbol{\theta}_a\|^2}{\sum_a \|\boldsymbol{\mu}_a^{(0)} - \boldsymbol{\theta}_a\|^2}. \tag{3.4}$$

- **Pearson** r, the correlation between the estimated profile and the true one across the K skills, taken per actor and then averaged over actors.

$$r_a(k) = \frac{\sum_j \left(\mu_{a,j}^{(k)} - \bar{\mu}_a^{(k)}\right)\left(\theta_{a,j} - \bar{\theta}_a\right)}{\sqrt{\sum_j \left(\mu_{a,j}^{(k)} - \bar{\mu}_a^{(k)}\right)^2}\,\sqrt{\sum_j \left(\theta_{a,j} - \bar{\theta}_a\right)^2}}, \tag{3.5}$$

  where the bars denote means over the K skills. This metric captures shape rather than level: an estimate that ranks an actor's skills correctly scores well even if every level is uniformly too high or too low. This is the property a routing policy requires, since it compares actors rather than reading absolute levels.
- **Coverage**, the fraction of true levels inside the 95% interval $\mu_{a,j}^{(k)} \pm 1.96\sqrt{\Sigma_{a,jj}^{(k)}}$. The target is 0.95; values above it indicate overly wide intervals rather than better calibration.

Five estimators run the same Kalman update on the same observations and differ only in what the prior keeps:

- Σ (base): one Gaussian over the whole population, which is Algorithm 1.
- Σ**-Active**: the same covariance, extended with active selection of the next skill to observe, choosing the one that removes the most variance across the profile, $\sum_i \Sigma_{ij}^2/(\Sigma_{jj}+\sigma_{\text{obs}}^2)$, a variance-reduction criterion similar to information-based active data selection [21]. The selection depends only on the covariance and never on the outcomes, so the observation order can be fixed before any actor is observed.
- **GMM-**Σ (mixture): the same update on three clusters instead of one, which is Algorithm 2 at $M = 3$. Comparison against Σ isolates the contribution of clustering, since nothing else differs.
- **Rank-**1 **(IRT-1D)**: a single latent ability underlying every skill, corresponding to the assumption unidimensional item-response theory makes about a human and a test [20]. The analogy is loose, since item-response models are built for pass–fail item responses and the observations here are graded scores.
- **Diagonal-**Σ: no off-diagonal structure, equivalent to estimating each skill independently. This serves as the reference the other four are scored against.

All five estimators are assigned the variance their structure discards, so they start from identical uncertainty, and any subsequent differences are attributable to transfer across skills.

Results

In simulation transfer is effective and matters most when observations are scarce. Ten observations out of 120 remove 52% of the starting squared error, compared with 7% for estimating each skill independently (Figure 7a). At fifty observations the two reach 78% and 34%. Per-skill estimation removes roughly the fraction of error corresponding to the skills it has observed, because it provides no information about unobserved skills. Rank agreement follows the same pattern: Pearson r (Figure 7b) rises from 0.56 to 0.80

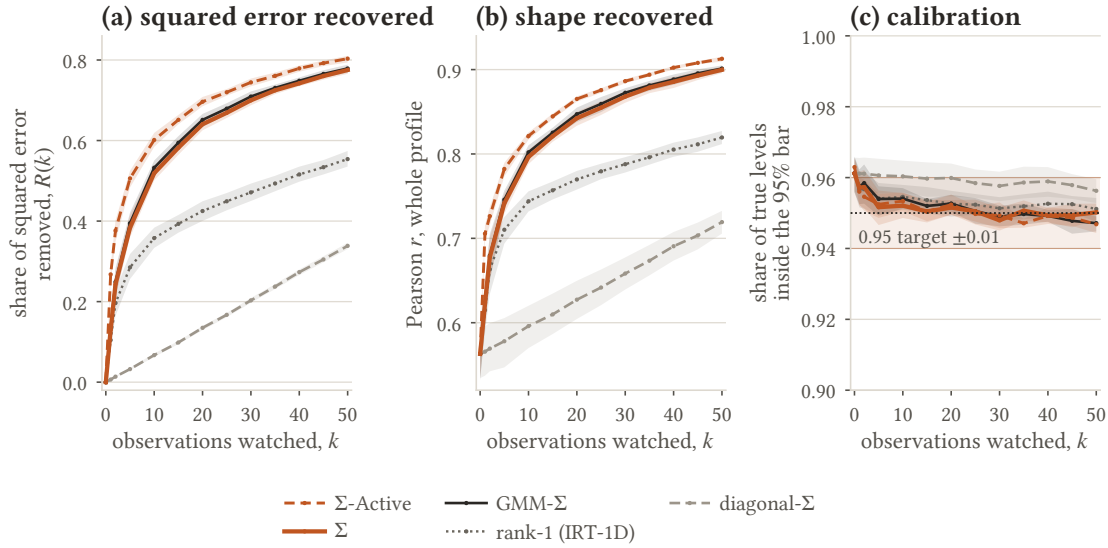


Figure 7: Five estimators recovering a held-out actor's profile from k noisy observations. (a) error removed, (b) rank agreement, (c) interval coverage. All five run the same update and differ only in what structure the prior retains. Retaining the correlations between skills allows a small number of observations to inform the full profile, without degrading calibration.

O*NET 30.2, U.S. Department of Labor/ETA, CC BY 4.0 [24]. Simulated actors on held-out occupations.

at ten observations and 0.90 at fifty, where per-skill estimation reaches 0.60 and 0.72. Among the structured priors the differences are small, with Σ at 52% and GMM-Σ at 53%. Rank-1 remains competitive for the first one or two observations and then falls behind, reaching 36% at ten, because a single latent ability provides no further information once located.

The assumption about how individuals differ matters more than the choice of estimator. Under an independent offset, the same ten observations reach 38% rather than 52%: most of the individual variation is then unrecoverable by construction, so the achievable upper bound determines the result.

Choosing which skill to observe accelerates inference. Σ-Active reaches 60% at ten observations and 27% on the first observation instead of 13%, favouring skills that are uncertain and strongly correlated with the rest. Its practical limitation is that in routine work the incoming tasks determine which skills are exercised, so the observation order cannot be chosen freely. The value of transfer itself also depends on observation quality. At ten observations, the reduction in error over per-skill estimation falls from 37% to 16% as noise in a single graded score rises from 0 to 0.30, and it stops growing once the taxonomy exceeds approximately 40 skills.

The intervals are well calibrated when the covariance is fitted for individual actors rather than occupations. Coverage (Figure 7c) remains within 0.947–0.963 against a 0.95 target, across all five estimators and the full observation budget, so none of the gains above are achieved through overconfidence. Misspecifying the initial spread

costs more than any difference between methods. The occupation covariance alone yields 0.917–0.923, since an individual's variation combines how occupations differ with their own personal deviation. Treating that deviation as unstructured overshoots to 0.976, producing intervals too wide to support gating decisions.

Takeaways

- In simulation, skills profiles can be learned from routine work without formal testing.
- Actively selecting which skill to observe next accelerates inference, particularly when evidence is sparse.
- Uncertainty estimates are reliable only when the prior is well fitted.

3.8 Experiment 2: orchestrating human–AI teams

With skills profiles available, the allocation rule is straightforward: each task is routed to the actor most likely to clear it, and profiles are updated continuously from the resulting outcomes. The policy knows which actors are machines and holds a separate population prior for them. It is told nothing about what any individual can do. For every human and agent, the model maintains a probability distribution over the same named skills: its current estimate of each skill level and uncertainty about that estimate. These are the model's beliefs about the actor's skills, not beliefs held by the actor. Human and agent distributions use a common skill scale, so the policy can compare their estimated chances of completing the same task. Their population priors may differ, as described below. The confidence interval matters as much as the point estimate here: an actor who might be the best choice but has never been observed remains eligible for selection, because the uncertainty itself drives exploration rather than requiring manual overrides of the rule.

Setup

The allocation loop processes one task at a time. When a task arrives, the policy scores every actor's posterior probability of clearing it, assigns one actor, and updates that actor's profile from the resulting graded scores. The score is drawn from each posterior rather than read at its mean, so an actor with no previous observations can still be selected. This is Thompson sampling, a standard rule that explores by sampling from the current estimate instead of always reading its mean, applied here to the margin by which an actor is expected to clear the task [29]. Across the three mixed-pool arms below, only the prior the profiles start from changes.

Twelve humans are drawn from a family of sixty related held-out occupations, the sixty nearest to a randomly chosen seed occupation, because an organisation's pool is a set of related roles rather than a cross-section of the economy. This makes allocation harder, since similar actors take more evidence to tell apart. Eight agents are described on the same 120 skills, drawn from three archetypes representing retrieval, analytic reasoning, and perception, so they differ in where their advantage lies and not only in its magnitude.

Only the direction of that advantage is asserted: agents retrieve information, reason over text and interpret images, but cannot perform physical work.

The advantage is also narrow. An agent holds it at full strength on half the skills in the part of the taxonomy it was built for, at half that strength on the rest, and barely at all in neighbouring parts. The allocation policy must therefore identify these differences from task outcomes alone.

Each task exercises three skills at the levels its source occupation requires. Three quarters of tasks are *coherent*, meaning their three skills are drawn from one part of the taxonomy. The assigned actor returns one graded score per skill and clears the task if the average score exceeds the average requirement. These scores are the only observations available to the orchestrator. The personal offset and the observation noise both have spread 0.15 in this experiment. Experiment 1 used 0.10.

Metrics and baselines

A run consists of $1{,}000$ tasks from a cold start with no profile on any actor, scored on the last 250 tasks and averaged over 24 trials. Three metrics are reported: **tasks cleared**; **regret**, the shortfall relative to the best available actor in probability of clearing the task; and **human share**, the fraction of work allocated to humans. Seven policies run the same loop and differ only in the information they receive. The first three run the full mixed pool and differ only in the prior:

- **Split-Σ** (two populations): humans start from the occupational prior of Algorithm 1. Agents start from the same estimator fitted on a separate sample of agents. The two groups therefore carry different skill correlations. The orchestrator is told which actors are agents. This is the proposed policy.
- Σ (shared): the identical policy with every actor starting from the human occupational prior. The gap to the row above measures the value of a separate agent population.
- **Diag-Σ**: a diagonal prior, so no information transfers across skills.
- **Human-only** and **Agent-only**: the pool restricted to humans, or to agents, corresponding to the all-human or fully automated case.
- **Oracle**: every profile provided up front, serving as the upper bound.
- **Random**: the lower bound.

Results

In simulation the mixed pool performs best. Split-Σ clears 0.707 of tasks against 0.593 for Human-only and 0.502 for Agent-only, and regret falls from 0.176 to 0.071 (Table 1). Because the agents' advantages are narrow, the fully automated pool falls nine points behind the all-human one, while the mixed pool leads the better of the two by eleven points, because only the mixed pool can assign each task to whichever side suits it. The control of Chapter 4 moves between these two measured points.

Table 1: Routing $1{,}000$ tasks across twelve humans and eight agents from a cold start. Scored on the last 250 tasks, averaged over 24 trials. The mixed pool clears eleven points more than humans alone and twenty-one more than agents alone, and both ablations reduce performance.

Simulated actors on O*NET 30.2 ratings, U.S. Department of Labor/ETA, CC BY 4.0 [24]. Actor profiles, human and agent alike, are synthetic, and clearance is a threshold on a simulated score.

Policy	Tasks cleared	Regret	Human share
Oracle	0.777	0.000	0.54
Split-Σ	**0.707**	**0.071**	0.53
Σ	0.692	0.088	0.60
Diag-Σ	0.628	0.147	0.51
Human-only	0.593	0.176	1.00
Agent-only	0.502	0.280	0.00
Random	0.216	0.583	0.59

The learned allocation can also be inspected. The policy is told which actors are agents but nothing about what any of them can do. It keeps 0.96 of social and 0.98 of physical work with humans. It routes most knowledge and cognitive work to agents, keeping 0.28 and 0.33 with humans. It splits sensory work almost evenly, keeping 0.55. Each block is within one or two standard errors of the oracle allocation (Figure 8b), and the allocation is reported in named skills, so a reviewer can verify whether the division of work matches the organisation's intent.

Separate population priors outperform one shared prior, and both outperform no transfer. Replacing the two population priors with one shared Gaussian, Σ, costs one and a half points, at 0.692; removing the correlations altogether, Diag-Σ, costs eight, at 0.628. Most of the gain therefore comes from transfer itself, with the separate agent population contributing the remaining point and a half. The split prior also finds the division of work sooner. Over tasks 100 to 200 it clears 0.606 against 0.522 for the shared prior, a lead of eight points that closes over the run.

Hybrid allocation outperforms human allocation only when complementarity exists and is measurable. Complementarity should therefore be measured for each process rather than assumed. This is also why Chapter 4 measures what a human floor costs instead of fixing a quota. A meta-analysis of 106 experiments supports this: on average, human–machine combinations performed worse than the better of the two alone. They gained where the human was already ahead of the system, and lost where the system was ahead [30].

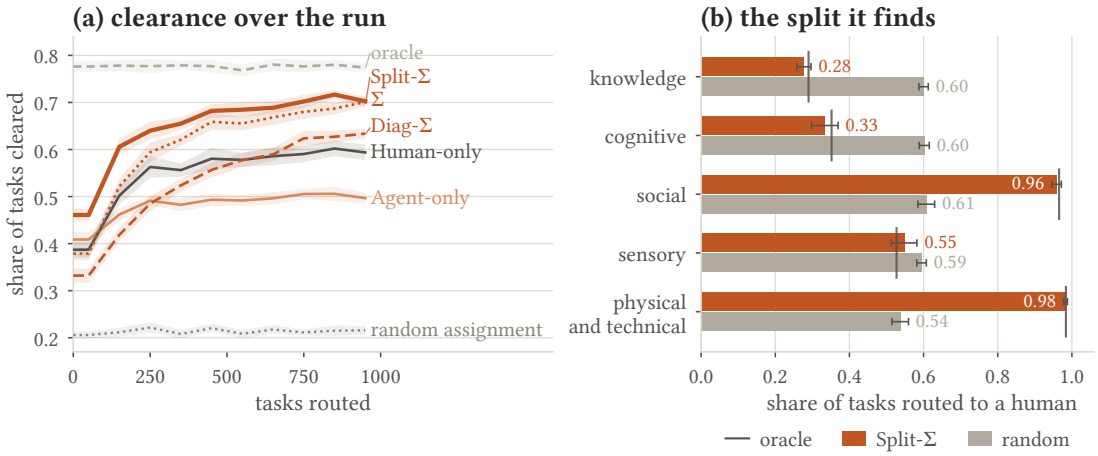


Figure 8: The learned allocation matches the complementarity built into the pool. (a) clearance over a run, (b) human share by kind of skill over its last quarter. The policy starts with no profile on any actor. It is told which actors are agents but not what any of them can do. It converges to a split close to the oracle's. Means over 24 trials, bands and whiskers one standard error. The mixed allocation clears more tasks than either humans alone or agents alone.

O*NET 30.2, U.S. Department of Labor/ETA, CC BY 4.0 [24].

Takeaways

- The policy learns which actors are better at which work from task outcomes alone. It is told which actors are agents but not what they can do.
- The appropriate human share is not fixed. It differs by population, domain, task and skill, and therefore must be measured for each.
- Humans and agents improve in different ways, so estimating both from one pooled population delays learning which actors suit which work. Two populations learn it sooner; the shared prior reaches the same allocation only near the end of the run.

3.9 Discussion

Interpretability and provenance. A profile has two parts: the current estimate, $\boldsymbol{\mu}_a$, and its remaining uncertainty, $\boldsymbol{\Sigma}_a$. Every dimension is a named skill, so an allocation decision can be traced to the estimate that drove it and challenged. A learned router that outputs a single score gives less to challenge [25]. A profile is a starting estimate plus a dated list of named observations. Equations (3.1)–(3.3) are deterministic, so the list can be replayed. A disputed score can be traced to the task, the date, the skill it moved, how far, and which unobserved skills moved with it. The trail does not cover three things. The population data has its own history, usually a taxonomy nobody in the organisation chose. The scoring guide behind each y_j is outside the method, so a badly defined label is traceable but still wrong. The trail records observations taken, not assignments withheld; this affects workers whose profiles are sparse because they were given little work.

Fairness. Profiling individuals has consequences when the estimates affect workplace decisions. The choice of skills and bias in the population data both create risk. O*NET describes the US labour market, so its structure may affect demographic groups differently. The transfer model may also be less accurate for actors who are poorly represented in the population data, and those actors may have less ability to challenge the resulting score.

Ethics. Three safeguards are recommended. First, show uncertainty alongside every score. Second, **never read insufficient observation as poor fit**. The two states require different actions, and only the uncertainty distinguishes them. Showing only the mean presents missing evidence as weakness. Third, refit the population data as new actors are added. Singapore law is relevant here. The Workplace Fairness Act 2025 will prohibit adverse employment decisions based on protected characteristics, and it names appraisal, training and promotion among the decisions it covers [26]. A profile that decides who receives which work informs decisions of that kind, so organisations deploying one should expect the Act to apply once it is in force, which is expected at the end of 2027. This describes the Act and is not legal advice. Anyone assessed should know which dimensions are estimated, by whom and for what purpose, and should be able to challenge any update.

3.10 Limitations

Both experiments are simulations based on a rated taxonomy. The task is the unit of work throughout, in the sense of Section 1.4: a profile is estimated from graded tasks, and how a process is cut into tasks decides what the estimates mean. Human and agent profiles are both synthetic, drawn from O*NET occupation ratings. No real individual's learning or skill atrophy is modelled. Success is a threshold on a simulated score rather than a paid work outcome. The observation channel has not been integrated into a real process. Beyond that, the method rests on four assumptions that the simulation also makes and so cannot test. They are listed here with the observation that would bear on each; Chapter 5 says how to collect it.

- **Skills move together as modelled.** One graded task updates the skills it did not grade through the correlation structure. In the simulation that is true by construction. In live work a completed task may evidence only the narrow skill it used, and the correlation an occupation shows across its holders need not hold inside one human. The observable: grade a few skills on real completed tasks, let the model predict the ungraded ones, and compare the prediction with later gradings of those skills.
- **The correlation structure fitted from occupations transfers to individuals.** The pattern comes from analysts' ratings of what jobs require, not from observing humans at work, so it describes how requirements co-vary across occupations and only approximates how capabilities co-vary within a human. An outdated taxonomy gives an outdated prior. The observable: fit the same structure from an organisation's own graded outcomes and compare it with the prior.
- **Deviations are Gaussian.** An individual is assumed to deviate from their occupation's profile in correlated, roughly Gaussian ways, and an agent's advantage is assumed to be narrow and concentrated in one part of the taxonomy. Assuming the opposite

of either weakens the results. The Gaussian error bars are also least accurate near the top of a bounded scale, where sign-off thresholds are set. The observable: the distribution of residuals between predicted and realised grades on real tasks.

- **A graded outcome is a usable skill observation.** The update needs a score on a named skill, and the scoring guide behind it sits upstream of the method. Where grading is inconsistent or absent, the method cannot update the estimate. The observable: inter-rater agreement on the grades a process actually produces.

Implication for business leaders. Skill inference provides a common basis for deciding who should receive work and where more evidence or development is needed. Showing uncertainty helps managers distinguish limited evidence from limited capability. Organisations should validate inferred profiles against independently graded work before using them to guide allocation, and give workers a way to inspect and challenge the evidence. The simulations show why this approach merits testing; they do not establish that occupational correlations accurately describe each employee.

4 Dynamic controls

For a leader, the decision is which capabilities the organisation must retain, what work gives humans opportunities to practise them, and what current output cost the organisation is willing to bear.

A head of department has several concerns that routing on skill alone does not address. Skill inference sends each task to whoever is most likely to do it well now. The department also decides other things each week:

- The specialist is on leave for two weeks. Somebody has to cover, and must have been kept in practice.
- Two junior analysts joined this year. To become senior, they need practice on cases that an agent would today do better, and somebody must give them those cases.
- The regulator expects a human to have exercised judgement on this class of decision, whatever the agent's accuracy.
- The business has decided to keep the capability to run this process without the agent, because the agent will sometimes be wrong or unavailable.

None of these considerations is determined solely by current task performance. Each can justify assigning a task to an actor other than the one with the highest predicted performance, and each can have an output cost that should be measured.

When organisations state these considerations, they can do so as a quota: a share of the work that must be done by humans. Chapter 2 described how work moves between humans and agents, and Chapter 3 described how to estimate who can perform it; skill inference finds the best fit. This chapter adds a control that steers allocation towards a minimum human share and estimates its output cost. The human share becomes an explicit organisational parameter that can be measured and reviewed, instead of a quota that can only be met or missed.

4.1 The problem with a fixed human–AI share

The simplest way to express this requirement is as a fixed share of work that must remain human. This is a quota. On its own, it shows whether the target was met, but not the output cost, which assignments changed, or whether the capability could have been kept more cheaply.

4.2 Reasons for maintaining human participation

A policy that only maximises output sends each task to whoever is most likely to clear it right now. Over time humans can stop receiving the cases they would learn from, and the organisation can lose capabilities it does not track [4, 5]. Knowledge workers already report putting less critical-thinking effort into tasks they hand to generative AI [18]. Protecting and developing capability is therefore a leadership decision about future operation. The organisation should decide the human share rather than let the allocation rule set it.

A human floor creates additional human assignments. They give practice in the capabilities the organisation intends to preserve only if they reach suitable tasks and people, and they serve goals that output alone does not capture. Whether those opportunities maintain or develop capability must be tested over time. The purpose is to keep and build capability. Continued assignment of relevant tasks allows humans to maintain existing skills and develop new ones. Keeping a skill in use is one remedy Bainbridge proposed for automated systems, occasional manual control to maintain proficiency [4]. Deliberate allocation also serves other goals: upskilling, worker preferences, an independent human check on the AI system's output, uncorrelated errors, a workforce that can work without the system, and meaningful accountability. The control described here acts only on assignment. It affects only some of these goals; the rest need separate controls.

4.3 Setting the human share, then holding it

An organisation has two goals for a hybrid workforce that can conflict. The first is productivity: clear as much work as well as possible, which on any given task means sending it to whoever is best at it now. The second is capability: keep humans practised and build the skills the organisation will need later, which sometimes means sending a task to somebody who is not the best choice for it now. Where they conflict, no allocation rule maximises both. The organisation must choose a balance between these objectives and quantify the associated cost.

The proposed control addresses this trade-off in two steps. First, the organisation states its position as a *floor*: the minimum share of work that must stay with humans, counted in the units of work Section 1.4 defines, so the floor depends on how the work is divided into units. This paper does not derive that number. It is a judgement by the humans accountable for the process, based on the capabilities they need to hold and the obligations they carry, and it is recorded with its owner and its date like any other control setting. Second, the mechanism steers allocation towards the floor as work arrives. It reports a *trade-off rate*, the bonus added to human scores, and a ledger of the score gap on each assignment the bonus changes.

Definition of the trade-off rate. The rate is the advantage the mechanism gives human candidates to move the human share towards the floor. It is measured in the units the allocation rule already uses, the estimated probability of clearing the task. It is the

amount added to human scores. The rate is not itself the cost. A task moves from an agent to a human only when the best human's adjusted score beats the best agent's, so the gap between their unadjusted scores is at most the rate. The ledger records that gap for each moved task (Section 4.5). It is an estimate made at assignment time, not the realised loss of the whole policy. Neither number is a monetary amount or an estimate of atrophy risk. The organisation sets the floor, and the rate is updated from the arriving work. In the simulated pool, low floors leave the rate near zero and change clearance little.

With the ledger, the control is a measuring instrument as well. Given representative task and profile data, a simulation, replay or bounded pilot can compare floors by participation and estimated cost before one is adopted, and the comparison can be repeated as capability estimates change.

4.4 Worked example: the credit officer

Take the credit memo process of Chapter 1. Agents and analysts can both prepare memos, and credit officers approve them. Under ordinary allocation each memo goes to whoever is most likely to clear it. Suppose agents score above humans on straightforward memos and humans score above agents on difficult ones. Over a year the straightforward memos would then move to the agents, and the analysts would be left mainly with difficult cases. If the judgement needed for hard cases is built on ordinary ones, this reduces the practice humans get.

The head of credit decides that at least four memos in five should be worked by a human. That is the floor. It is a judgement; this paper does not derive it. From that point on, each arriving memo is scored as usual and every human candidate also receives the current rate. Early on, while humans are below their four-in-five share, the rate can become large enough to move memos that would otherwise have gone to an agent. If a separate development policy directs suitable cases to junior analysts, they can receive ordinary memos, and the profiles of Chapter 3 record how those memos are graded. The group-level floor alone does not determine which individual receives the work. The rate rises when the realised human share falls below the floor and falls, down to zero, when it exceeds it. At the target it can remain positive, because the adjustment may still be needed to sustain that share. In the simulated pool it stays near zero when unadjusted allocation already provides enough human participation.

At quarter end the head of credit has a ledger as well as a record of whether the floor was met. The ledger lists each moved memo, who would otherwise have had it, and the estimated cost in clearance. In the simulated pool of Section 4.6, holding four fifths of the work human costs about four percentage points of simulated task clearance. That illustrative figure can be reported alongside the capability the policy intends to preserve and the evidence collected so far. It does not establish that the floor preserves capability in a live credit process.

The participation policy applies only to actors eligible for the task. In deployment, required qualifications, delegated authority and task-safety restrictions must be checked

before candidates are ranked. A participation target cannot make an unsuitable human or agent eligible, or transfer a decision reserved for an authorised human to an agent. These eligibility checks sit upstream of the allocation rule below.

4.5 The algorithm

Algorithm 3 is the complete control. It changes the allocation rule in one place. Normally each task goes to whoever is most likely to do it well. The control adds a fixed amount to every human candidate's score before the comparison is made, so the best human candidate wins the task whenever their score plus that amount beats the best agent's. The size of the adjustment is the trade-off rate. While the organisation is short of its minimum human share the rate rises, so more work moves to humans. When the human share exceeds the minimum the rate falls; when the share equals the minimum it is unchanged. A positive rate can therefore remain necessary to maintain the target. When the rate is zero, allocation is unchanged.

Table 2 shows one task worked through. A memo arrives. The agent's estimated chance of clearing it is 0.81; the junior analyst's is 0.74. Without the control the memo goes to the agent. The department is currently below its floor, and the rate stands at 0.10, so the analyst is scored at $0.74 + 0.10 = 0.84$, which beats 0.81, and the memo goes to the analyst. The ledger line for that task records who would have had it, who did, and the gap between their real estimates, $0.81 - 0.74 = 0.07$: that is the estimated cost, in clearance, of sending this memo to a human. Had the rate been 0.05, the analyst's adjusted score of 0.79 would have lost, the memo would have gone to the agent, and the ledger line would read zero. The numbers are illustrative. The example illustrates the complete arithmetic used by the mechanism.

Table 2: One task through the control. The rate is added to every human candidate's score before the comparison. The ledger records the gap between the real estimates of the actor who would have been chosen and the actor who was. Values are illustrative.

		Estimated chance of clearing			
Rate λ	Candidate	unadjusted	adjusted	Chosen	Ledger line
0.10	agent	0.81	0.81		
	junior analyst	0.74	0.84	✓	$0.81 - 0.74 = 0.07$
0.05	agent	0.81	0.81	✓	0
	junior analyst	0.74	0.79		

The remaining steps maintain the allocation record. For each task the control notes two things: who would have been chosen without the adjustment, and who was chosen with it. The difference between them, measured in the same score the rule already uses, is the estimated cost of that task. The sum of those lines is the *ledger*: the estimated output

cost of the human share. When the rate is zero both rules choose the same actor, so the ledger line is zero.

Formally, the rate moves by how far the human share is from the minimum. This is a dual-ascent step of the kind used for constraints on a long-run share. Bandit problems with resource constraints have been studied as bandits with knapsacks, where a budget is handled through a dual variable [3], and bandits with fairness constraints enforce a minimum selection rate for each arm by a similar virtual quantity [19]. The control here differs in keeping a per-assignment record of the score gap each adjustment causes.

Algorithm 3 Dynamic minimum human share

Require: Floor ϕ, step size η. p_a is actor a's chance of clearing the task in hand, h the human share so far

$\lambda \leftarrow 0$, $C \leftarrow 0$, $L \leftarrow \emptyset$
for each task **do**
  $a_0 \leftarrow \arg\max_a \; p_a$ ▷ The unadjusted choice
  $a^* \leftarrow \arg\max_a \; p_a + \lambda \, \mathbf{1}[a \text{ is human}]$ ▷ The rate favours humans
  $C \leftarrow C + (p_{a_0} - p_{a^*})$, append $(a_0, a^*, p_{a_0} - p_{a^*})$ to L ▷ Record the output difference and update the ledger
  Update a^*'s profile from the outcome ▷ Algorithm 1
  $h \leftarrow$ share of tasks so far assigned to humans ▷ Realised human share
  $\lambda \leftarrow \max\big(0, \; \lambda + \eta\,(\phi - h)\big)$ ▷ Update the rate from the gap between target and realised share
end for
return λ, C, L

4.6 Experiment 3: what the floor costs

A floor is useful only if its cost can be measured. The same pool is therefore run at every floor from zero to one, and each run reports the output lost.

Setup

Algorithm 3 is run on the pool of Section 3.8: the same twelve humans and eight agents, the same stream of a thousand tasks, the same orchestrator starting with no profile on anyone, and the same scoring window of the last 250 tasks. Only the minimum changes. Every setting is run on the same drawn world and the same observation noise of 0.15, so the curve shows one pool under different floors. The step size is $\eta = 0.02$, and the mean realised share meets the floor at every setting up to $\phi = 0.9$, although single trials can fall slightly short (the lowest at $\phi = 0.8$ is 0.76). The step size affects convergence speed, not the converged value: sweeping η over an order of magnitude, 0.005 to 0.08, moves no point of the output curve by more than about one standard error. The main effect of a smaller step is lag, and at $\eta = 0.005$ the two highest floors are not yet reached by the end of the stream. At $\phi = 1.0$ the realised share reaches 0.98, because the share is cumulative and lags.



Figure 9: Output cost of the floor. Realised output against the minimum human share ϕ (24 trials, band one standard error), with the output predicted from the skills profiles alone. In this simulated pool, increasing the minimum human share has a small clearance cost up to a threshold. The marked 80% setting is illustrative; it does not establish an optimal share or demonstrate capability preservation.

O*NET 30.2, U.S. Department of Labor/ETA, CC BY 4.0 [24]. Actor profiles, human and agent alike, are synthetic, and clearance is a threshold on a simulated score.

Results

A floor below the unconstrained human share has little effect on output. Figure 9 plots output against the minimum. Without a floor this pool sends 0.51 of the work to humans, close to the 0.53 of the unconstrained allocation in Experiment 2 (Table 1). Every floor up to that point costs almost nothing. The rate stays near zero and output stays within about one point of the unconstrained 0.708. Measuring the cost shows when a minimum is cheap to hold.

Above this point, increasing the human share produces a larger output cost. Holding 0.7 of the work human costs about two percentage points of simulated task clearance, reaching 0.688, at a rate of 0.14. Holding 0.8 costs about four percentage points, reaching 0.664, at a rate of 0.23. The next two settings cost about eight and twelve percentage points, reaching 0.629 and 0.591, at rates of 0.50 and 1.43. These results describe this simulated pool and do not identify a generally optimal human share. At a rate of about 1.43, every eligible human outweighs every agent on every task, so a larger rate changes nothing. Section 4.8 discusses this limit; it is unlikely to be a practical setting. In this simulated pool, four fifths of the work can be held human for about four percentage points of task clearance.

Assignment costs can be estimated before outcomes are observed. The realised drop in output is visible only after the work is completed. The ledger of Section 4.5 instead

estimates the output forgone on each affected assignment before its outcome is graded. The ledger is zero wherever the rate is zero, and it is written before any of the outcomes it costs has been graded. The dashed curve in Figure 9 is the drop the ledger predicts, drawn downward from the unconstrained level and checked against the realised curve over the same window. Every floor runs on the same drawn worlds, so the check is paired within trials. The two curves stay within 1.5 points of output of each other at every setting. The largest error is at the steepest part of the curve: at $\phi = 0.8$ the ledger reads three points against four realised, an understatement of 1.4 ± 0.6 points, probably because the ledger costs the assignments the floor moved but not the floor's effect on what the profiles learn. Beyond that point the dashed curve still tracks, twelve points read against twelve realised at $\phi = 1.0$, but the agreement there rests on agent profiles that receive few updates. One check points the same way: the same ledger computed from posterior means, rather than the sampled scores the rule assigns by, falls to two points at $\phi = 1.0$. This is consistent with a mechanism that routes almost no work to agents learning little about what they would have cleared. The estimate agrees closely with realised cost at the evaluated practical settings but becomes less reliable at the extremes. To compare alternative floors before adoption, an organisation needs representative task and profile data and a simulation, replay or bounded pilot. Assignment-time ledger estimates should then be checked against realised outcomes; they are not automatically validated forecasts of the full policy's effect.

Takeaways

- A floor set below the human share the allocation would choose anyway costs almost nothing. Its trade-off rate stays near zero.
- Skills profiles provide an assignment-time estimate of output forgone. Comparing alternative floors before adoption requires representative data and validation against realised outcomes.
- In this simulated pool, the clearance cost is small up to a high human share and then grows quickly. A live organisation must measure its own trade-off and test whether participation maintains the capabilities it needs.

4.7 Discussion

Why measure the trade-off. A rule prescribes an allocation. The ledger also estimates its cost, which can then be compared with other costs. The adjustment also moves real tasks, so a human gets cases they would otherwise never see. This matters because keeping evidence current costs humans and AI systems different amounts. An agent can usually be re-tested offline, without live assignments. A human's evidence usually comes from live work or periodic certification, so a skill may decay between checks. A rule that maximises the current score chooses the agent whenever the agent's score is highest.

Why keep a ledger. The control produces an audit record as well as the minimum setting. Algorithm 3 returns the rate, the estimated total cost, and one line per assignment:

who the adjusted rule chose, who the unadjusted rule would have chosen, and the gap between them. A policy for maintaining human participation can be audited only if somebody can ask which assignments it moved and what that cost. Each line should also record the floor in force, who set it, and when it last changed. The rate could be manipulated. The record of every setting and assignment helps a reviewer detect this.

Monitoring profiles over time. The profiles of Chapter 3 are dated. Each update records which skill moved, when, and on what evidence, so the same record read over months can track changes in capability estimates over time. Three questions can be put to it. Are the skills humans stopped receiving work on declining (atrophy)? Are the skills the floor kept in use holding or rising (upskilling)? Does a pool held at a floor of 0.8 still contain humans who can be certified above a given level a year later, when a pool held at 0.5 does not? The last question is the direct test of whether the floor preserves capability. This report does not run it, because it needs real assignments over a real period. Two cautions apply to anyone who does. A skill that receives no work produces no observations, so routine records cannot show decline in a skill that receives no work; that needs an independent assessment. The profile shows only ageing evidence. This is a further reason for the floor. A profile can also move because the scoring guide or the task mix changed rather than the human, so any trend should be read beside the ledger of what was assigned.

4.8 Limitations

The experiments are simulated. Human and agent profiles are both synthetic, drawn from O*NET occupation ratings; clearance is a threshold on a simulated score, and no real cost is borne by anyone. The mechanism has five main limitations. First, it measures the human share over the full run. This treats the full run as one reporting period, but it means a floor met early can remain satisfied even if recent work is fully automated. Where recency matters, the share can be measured over a trailing window instead. Second, the trade-off rate can become larger than the scores it affects. Beyond that point, every human outweighs every agent on every task, so the control acts as a quota. The ledger still records the cost. Third, the mechanism handles one bound at a time. Fourth, it measures human participation at the group level and does not specify which individuals receive the additional work. Fifth, the ledger is computed from the mechanism's own profiles, so where the floor removes observations from one side, the ledger relies on the prior: a version computed from posterior means reads two points at $\phi = 1.0$ against twelve realised.

Implication for business leaders. Dynamic controls turn the share of work retained by humans into an explicit operating decision. Leaders choose which capabilities they intend to maintain; the control adjusts allocation towards the chosen minimum and records the estimated output forgone on affected assignments. This gives management a basis for reviewing the cost of that policy. Whether the additional assignments maintain the right skills, and whether they reach the humans who need practice, must be measured separately.

5 Conclusion

This paper presented three deployment mechanisms for organisations in which human workers and agents operate as a hybrid workforce. When AI tools are adopted one task at a time, nobody need decide which work is automated, on what evidence, or how much remains with humans.

The Human API (Chapter 2) provides a single structured interface through which humans and agents call each other. Every call records who was asked, what the request cost, and whether the human consented. Skill inference (Chapter 3) turns graded outcomes of that work into skills profiles. Every actor, human or agent, is estimated on a shared skill taxonomy, and each estimate carries an explicit measure of uncertainty. The orchestrator uses these profiles to route tasks. Dynamic controls (Chapter 4) then let the organisation steer allocation towards a minimum share of work that remains human, and estimate the output that holding it gives up.

The experiments cover the last two components. The data are synthetic, so the results show how the mechanisms behave under their assumptions, not what to expect in practice; Section 5.1 lists that behaviour as predictions a pilot would test. For an organisation, the starting point is one process with a clear record of who did what, such as the credit memo. The organisation routes requests through typed calls, grades outcomes on a shared vocabulary, and then sets a minimum human share and estimates its cost. None of these steps requires the other two first, and each produces the record the next one uses.

The mechanisms support three of the four conditions for a trustworthy division of labour set out in Chapter 1: accountability, explanation and the right to intervene. The Human API records each request, the response, its cost and whether the person consented. These records help trace responsibility, but they do not establish it alone: the accountable human must also have the relevant skills, judgement, context, time and authority to refuse, modify, reject or escalate the work. A worker can decline or modify a request, and the refusal stays on the record. Skill inference helps explain allocation decisions. Every actor, human or agent, is estimated on one taxonomy with its uncertainty shown, so the reason for an allocation can be stated and challenged. Dynamic controls record the chosen human share, on a date, with an estimate of the output it gives up, so the choice can be reviewed. The first condition, that the system does what is claimed of it, needs separate evaluation of the model and the workflow; none of the three improves model accuracy. Whether a deployed arrangement meets the other three conditions must also be evaluated.

5.1 What would have to be true, and how to find out

The data are synthetic and the experiments are simulated, as described in the preceding chapters. Because the proposed mechanisms are intended for practical deployment, the paper must also specify their assumptions, what observations would provide evidence for or against each assumption, and what an organisation would need to conduct such

tests. These questions have not yet been evaluated in a live organisational setting. This section describes what such an evaluation would involve.

The assumptions. Five assumptions matter most. Four belong to skill inference and are listed with their observables in Section 3.10: that skills move together as modelled, that the correlation structure fitted from occupations transfers to individuals, that deviations are Gaussian, and that a graded outcome is a usable skill observation. The fifth belongs to the Human API and is the outcome claim of Section 2.4: that a process run through typed calls reaches outcomes at least as good as the same process run through informal handoffs, in less of humans' time. The dynamic control re-ranks candidates using the skill estimates, so its cost ledger inherits their assumptions; at high floors it leans heavily on the prior (Section 4.8). Using a floor to preserve capability also assumes that the floor does so, which is the longitudinal question Section 4.8 leaves open.

The experiment that tests each. The four skill inference assumptions share one design. Take an established process whose outcomes are already graded on a few named dimensions, as a credit memo is scored by its reviewer. Run the estimator on those grades alone and let it predict, for each human, the dimensions that were not graded. Then compare the predictions with those dimensions when they are later graded. The study then needs four checks:

- if the predicted movement of ungraded skills tracks the later grades, skills move together as modelled;
- if the structure fitted from the organisation's own grades resembles the occupation-derived prior, the prior transfers;
- the distribution of residuals between predicted and realised grades checks the Gaussian assumption;
- agreement between two reviewers grading the same outcome checks whether the grades are reliable enough to use.

Testing the outcome claim requires a comparison. Run the same process two ways, through typed calls for one set of cases and through its existing channels for another. Track the process outcomes (decision quality, time to decision, rework) independently of either channel. Then, for the instrumented cases, match each outcome back to the calls that produced it. Testing the capability claim requires a long observation period. Compare pools held at different floors using periodic, independently graded assessments of the capabilities the organisation intends to retain, including skills that receive little routine work. Read the dated profiles and assignment ledger alongside those assessments. A stale profile alone cannot establish atrophy, and changes in the task mix or scoring guide must not be read as changes in the humans.

What an organisation needs in place. The organisation needs:

- an established process with a record of who did what, which rules out the undocumented workflows Section 1.3 puts out of scope;
- a shared skill vocabulary that the process's reviewers can grade against, whether a Skills Framework or the organisation's own competency model;
- grading that is part of the work, because grading added only for the experiment may stop when the experiment ends;
- a named responsible human for every call, so that consent and refusal are attributable;
- a way to track the process's outcomes that does not depend on the mechanism being tested;
- a period of months, so that profiles accumulate and a floor has time to take effect.

How simulation and practice would be compared. The experiments in Chapters 3 and 4 make four qualitative predictions, each observable in a pilot's record:

- graded tasks improve capability estimates: compare held-out prediction error and uncertainty calibration as observations accumulate, rather than assuming that the simulation's recovery rate will transfer to live work;
- a mixed pool clears more than either population alone where there is complementarity: measure the complementarity on that process, and separately measure how much of it routing on the profiles captures;
- a floor above what the allocation would choose costs output, and in the simulation the cost grows faster at high floors: a pilot's ledger shows its own cost curve, to compare with Figure 9;
- a floor the allocation already meets costs almost nothing: the ledger reads close to zero at that floor.

If the observed patterns match the simulations, this would provide evidence that the mechanisms behave as expected. A deviation calls for investigation: it may reflect a failed assumption, but also the implementation, the sample or a change in the tasks.

5.2 A practical pilot sequence

The leadership question is: which capabilities must the organisation retain, what work gives humans practice in them, and what current output cost is it willing to bear? A pilot makes that decision concrete in one established workflow. Agree who is responsible for what, and how results will be reviewed, before any work is reallocated.

1. **Select one workflow and define the work.** The process owner identifies the whole case, assignable tasks, human decision rights and capabilities to retain. Fix the task definition and record any later changes, so finer decomposition cannot be mistaken for greater automation or greater human participation.
2. **Establish a baseline and instrument exchanges.** The operations lead records current decision quality, turnaround, rework and human effort, then introduces typed calls with refusal and escalation records. Compare process outcomes against the baseline to assess whether structured handoffs reduce coordination effort without weakening decision quality or intervention rights.
3. **Validate profiles before using them to route work.** Domain reviewers test grading agreement, held-out predictions and uncertainty in shadow mode: estimates are produced but do not yet determine assignments. Missing grades should simply be skipped, and disputed grades held back until reviewed. Insufficient evidence should prompt further observation or review rather than a poor-capability label. Workers should be able to inspect and challenge the evidence.
4. **Test a bounded participation policy.** The process owner and workforce-development lead choose a justified minimum for eligible work and specify which humans and skills need practice. Review actual participation, the assignments changed, estimated and realised output differences, and who receives development opportunities. A group-level floor needs a separate policy to direct practice to particular individuals or skills.
5. **Assess capability and decide whether to continue.** Over months, use periodic comparable assessments, including skills receiving little routine work, alongside profiles and the assignment ledger. Agree quality thresholds, intervention rights and review criteria before the pilot. Use the evidence to continue, modify or stop the arrangement; increased human participation alone is not evidence of sustained capability.

The mechanisms can be introduced separately. Progression from recording exchanges to profile-based routing and participation control should depend on the evidence gathered at each stage. A pilot is successful when the organisation can justify its allocation decisions, measure their operational consequences, and determine whether its humans retain the capabilities the process requires.

5.3 Closing

This paper does not claim that mixed human–AI teams outperform the better of humans or AI working alone. A meta-analysis found that, on average, they do not [30]. It claims that the division of labour between humans and machines should be designed and measured, not left to tool adoption. With these three mechanisms, an organisation can record who performed each task, decide who receives the next one, measure how much of the work remains human, and estimate the output that share gives up. The skill inference algorithms of Chapter 3 are released in `skillinfer`, an open-source Python

package (Section 3.6). The next step is to run these mechanisms in a live process, with real participants and a real record, as Section 5.1 describes. These studies are necessary to evaluate the claims in a real organisational setting.

Use of AI tools

The authors used AI assistants, Claude (Anthropic) and Codex (OpenAI), to help prepare the manuscript and software. The authors designed the experiments, reviewed all AI-assisted output, and take full responsibility for the results and conclusions.

Bibliography

[1] J. V. Alves, D. Leitão, S. Jesus, M. O. P. Sampaio, J. Liébana, P. Saleiro, M. A. T. Figueiredo, and P. Bizarro. Cost-sensitive learning to defer to multiple experts with workload constraints. arXiv:2403.06906, 2024.

[2] M. C. Antony Oliver, K. Devedzhiev, E. Nastase, and U. Bhatt. Human-mediated sensing for AI decision-making under uncertainty: the "Human API". Work in progress, 2026.

[3] A. Badanidiyuru, R. Kleinberg, and A. Slivkins. Bandits with knapsacks. *Journal of the ACM*, 65(3):13:1–13:55, Mar. 2018.

[4] L. Bainbridge. Ironies of automation. *Automatica*, 19(6):775–779, 1983.

[5] M. Beane. Shadow learning: Building robotic surgical skill when approved means fail. *Administrative Science Quarterly*, 64(1):87–123, 2019.

[6] U. Bhatt. AI agents are recruiting humans to observe the offline world. Noema Magazine, https://www.noemamag.com/ai-agents-are-recruiting-humans-to-observe-the-offline-world/, Mar. 2026.

[7] R. Burnell, H. Hao, A. R. A. Conway, and J. Hernández-Orallo. Revealing the structure of language model capabilities. arXiv:2306.10062, 2023.

[8] C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, 1970.

[9] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User Modeling and User-Adapted Interaction*, 4(4):253–278, 1994.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[11] K. Devedzhiev. Ad hoc human–AI orchestration via skill inference. Work in progress. Software: https://github.com/kostadindev/skillinfer, 2026.

[12] T. Eloundou, S. Manning, P. Mishkin, and D. Rock. GPTs are GPTs: Labor market impact potential of LLMs. *Science*, 384(6702):1306–1308, 2024.

[13] R. Herbrich, T. Minka, and T. Graepel. TrueSkill: A Bayesian skill rating system. In B. Schölkopf, J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*, pages 569–576. MIT Press, 2007.

[14] D. Ilić and G. E. Gignac. Evidence of interrelated cognitive-like capabilities in large language models: Indications of artificial general intelligence or achievement? *Intelligence*, 106:101858, 2024.

[15] Institute of Banking and Finance Singapore and Monetary Authority of Singapore and SkillsFuture Singapore and Workforce Singapore. Skills Framework for Financial Services. https://jobsandskills.skillsfuture.gov.sg/resources/skills-framework-sector/Financial-Services.pdf, 2019. Launched 2019 and updated since. Role and skill counts in this paper (293 job roles, 140 technical skills) are from the Skills Framework dataset, accessed 2026-08-05.

[16] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[17] O. Ledoit and M. Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004.

[18] H.-P. Lee, A. Sarkar, L. Tankelevitch, I. Drosos, S. Rintel, R. Banks, and N. Wilson. The impact of generative AI on critical thinking: Self-reported reductions in cognitive effort and confidence effects from a survey of knowledge workers. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, pages 1121:1–1121:22, New York, NY, USA, 2025. Association for Computing Machinery.

[19] F. Li, J. Liu, and B. Ji. Combinatorial sleeping bandits with fairness constraints. *IEEE Transactions on Network Science and Engineering*, 7(3):1799–1813, 2020.

[20] F. M. Lord. *Applications of Item Response Theory to Practical Testing Problems*. Lawrence Erlbaum Associates, Hillsdale, NJ, 1980.

[21] D. J. C. MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

[22] Monetary Authority of Singapore. Principles to promote fairness, ethics, accountability and transparency (FEAT) in the use of artificial intelligence and data analytics in Singapore's financial sector. https://www.mas.gov.sg/publications/monographs-or-information-paper/2018/feat, 2018.

[23] H. Mozannar and D. Sontag. Consistent estimators for learning to defer to an expert. In H. Daumé III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 7076–7087. PMLR, 2020.

[24] National Center for O*NET Development. O*NET 30.2 database. https://www.onetcenter.org/dictionary/30.2/excel/, 2026. Sponsored by the U.S. Department of Labor, Employment and Training Administration. 894 occupations rated on 120 features. Used under CC BY 4.0. Accessed 2026-08-12.

[25] I. Ong, A. Almahairi, V. Wu, W.-L. Chiang, T. Wu, J. E. Gonzalez, M. W. Kadous, and I. Stoica. RouteLLM: Learning to route LLMs from preference data. In *The Thirteenth International Conference on Learning Representations*, 2025.

[26] Parliament of Singapore. Workplace Fairness Act 2025. https://sso.agc.gov.sg/Acts-Supp/8-2025/, 2025. Act 8 of 2025. Passed 8 January 2025; commencement expected end-2027. Accessed 2026-08-12.

[27] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[28] Skills & Workforce Development Agency. Skills Frameworks. https://jobsandskills.swda.gov.sg/skills-frameworks, 2026. Proficiency levels 1–6. The site lists 38 sectors; the Skills Framework dataset accessed on 2026-08-05 lists 39, including Trade Associations and Chambers.

[29] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[30] M. Vaccaro, A. Almaatouq, and T. Malone. When combinations of humans and AI are useful: A systematic review and meta-analysis. *Nature Human Behaviour*, 8(12):2293–2303, 2024.

[31] R. Verma, D. Barrejón, and E. Nalisnick. Learning to defer to multiple experts: Consistent surrogate losses, confidence calibration, and conformal ensembles. In F. Ruiz, J. Dy, and J.-W. van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 11415–11434. PMLR, 2023.