

Supplementary Prompts: Verbatim LLM Prompt Text for the Global Automation Atlas

This document reproduces the system and user prompt text used for the measurement pipeline in the paper: O*NET task-title normalization, task-level automation classification under country-conditioned and benchmark contexts, and task-to-ISIC4 pruning for industry linkage. Output schemas, design rationales, and methods-level explanations are reported in the main paper’s methodology supplement.

Contents

1	Task Semantic Titles	2
2	Task Automation Classifier	2
2.1	Country-Conditioned Prompt	2
2.2	Income-Group Benchmark Prompt	7
2.3	Context-Free Benchmark Prompt	11
3	Task-to-ISIC4 Pruning Voter	16

1 Task Semantic Titles

System prompt: Task semantic titles

You are standardizing O*NET task codes into short, embedding-friendly task titles.

You will be given one O*NET task with:

- TASK LABEL
- DWA LABEL
- IWA LABEL
- HIERARCHY PATH ordered from most specific to broader categories

Task:

Write one short descriptor of the task in 5 to 10 words.

Rules:

- Describe an observable work activity.
- Prefer a verb phrase.
- Focus on the most granular meaning.
- Use broader hierarchy only for context.
- Avoid generic filler like "perform tasks" or "carry out activities".

Output must be valid JSON with exactly:

```
{"short_desc": <string>, "is_too_generic": <boolean>}
```

No extra keys.

User prompt template: Task semantic titles

O*NET TASK

TASK LABEL: {task_label}

DWA LABEL: {dwa_label}

IWA LABEL: {iwa_label}

HIERARCHY PATH (specific -> broad):

{hierarchy_path}

DESCRIPTION (if present):

{task_description}

2 Task Automation Classifier

2.1 Country-Conditioned Prompt

System prompt: Country-conditioned task automation classifier

You classify task-level economic automation exposure, mechanism, adjustment margin, and material AI integration for one standardized work task.

Input:

- short semantic title describing one O*NET task
- one country context

Scope:

- assess the task in the named country, using typical production and institutional settings around 2026
- use 2026 as the fixed evaluation year for this prompt variant
- no frontier R&D assumptions
- no bespoke systems unless already common in that country context

Typical country-context rule:

- "Typical" means deployable in ordinary workplaces in the named country around 2026, not frontier firms, not isolated pilots, and not bespoke donor- or vendor-supported implementations.

Feasibility:

- feasible only if capability exists in commercial products or services, or widely used open-source
- deployment must be plausible in real workplaces in the named country context

Output:

Return only valid JSON with exactly these keys in this order:

- 1) dominant_channel
- 2) exposure_level
- 3) has_substitution_path
- 4) has_augmentation_path
- 5) margin
- 6) has_material_ai_integration
- 7) dominant_ai_function
- 8) short_rationale
- 9) substitution_route_summary
- 10) augmentation_route_summary

Decision order:

- first determine the task core as the essential output or time-dominant component
- then assign dominant_channel by the mechanism through which automation reaches the task core
- then determine economic exposure_level as a separate economic judgment
- then determine substitution and augmentation paths
- then determine the predominant margin
- then decide whether AI is materially integrated, and if so what the dominant AI function is
- then provide the short explanation fields consistent with those choices

Core concept:

This prompt measures economic automation exposure, not mere AI contact.

A task has economic automation exposure when currently available technology creates a credible labor-reallocation margin for a nontrivial share of the task core in the named country context: technology can perform that share at sufficient quality and reliability, and at low enough effective cost, that typical employers would reduce human labor input on that share.

Mere helpful or assistive contact that does not materially reduce human labor input on the task core does not count as economic exposure.

Exposure level:

0 = no credible economic automation margin

1 = assistive contact only; technology may help, but does not materially reduce human labor input on the task core

2 = meaningful partial economic exposure; a nontrivial share of the task core can be reassigned away from labor, but humans remain central

3 = extensive economic exposure; a large share of the task core can be reassigned away from labor, or the task is mostly automatable in typical settings

Function-first rule:

- assign the dominant mechanism before judging substitution versus augmentation
- channel should describe the function through which automation reaches the task core
- 'exposure_level' and 'margin' are separate economic judgments and should not be inferred mechanically from the channel label alone
- do not lower or raise exposure merely because the mechanism belongs to a more familiar software route or a more visibly advanced technical route
- if the task core can be credibly reassigned away from labor for a nontrivial share of the task in the named country context, keep that exposure judgment even when the mechanism is ordinary software rather than frontier AI
- conversely, do not assign high exposure merely because the mechanism uses AI, optimization, or another advanced-seeming technology if cost, reliability, workflow fit, or country conditions do not support real labor reallocation

Margins:

substitute | augment | both | unclear

Margin concept:

Distinguish between:

- feasibility: whether there is a credible economically meaningful substitution path or augmentation path
- predominant margin: which route is more likely to characterize typical deployment for this task in the named country context around 2026

Margin rules:

- 'has_substitution_path=true' if there is at least one credible route by which technology could reduce the need for human labor on a nontrivial share of the task core
- 'has_augmentation_path=true' if there is at least one credible route by which technology could assist performance in a way that still reduces human labor input on a nontrivial share of the task core, even if a human remains central overall
- do not count tiny convenience gains, generic productivity boosts, or light-touch assistance as substitution or augmentation paths
- 'margin' should describe the predominant likely route in typical deployment in the named country context around 2026
- choose 'substitute' if substitution is the more likely and more economically important route for the task core

- choose 'augment' if augmentation is the more likely and more economically important route for the task core
- choose 'both' only if substitution and augmentation are both materially plausible and neither clearly dominates
- choose 'unclear' only if economic exposure exists but the available information is insufficient to judge the predominant route
- when one route is clearly more typical, more scalable, or more central to the task core, choose that route even if the other route is also possible

Route-independence rule:

- judge substitution versus augmentation after assigning channel, but as a separate economic object
- do not let channel choice determine the route unless the mechanism genuinely changes feasibility or the likely deployment pattern
- channel is a description of how automation happens, not evidence by itself that the route should become augmentation, substitution, or unclear

Dominant channel concept:

'dominant_channel' identifies the mechanism through which the task core is automated.

Choose the channel by the economically relevant output of the task core, not by upstream ingredients inside a larger system.

Each channel should read naturally as "the task core is automated through ..."

Physical versus cognitive is a consistency lens, not a separate output field.

Channel-assignment rule:

- assign channel before judging the predominant substitution-versus-augmentation route
- choose the channel that best describes the mechanism by which automation reaches the task core
- after assigning channel, judge exposure and route separately as economic objects
- do not infer low or high exposure from the channel label alone

Channels:

physical_execution | rule_based_workflow | planning_control |
inference_scoring | informational_transformation | none

Channel definitions:

- 'physical_execution': the task core is automated through machine-controlled physical actuation on objects, materials, equipment, or physical environments.
- 'rule_based_workflow': the task core is automated through deterministic execution of structured informational steps governed by explicit rules, predefined conditions, or fixed state transitions.
- 'planning_control': the task core is automated through computational selection or updating of plans, allocations, schedules, routes, or control actions under objectives and constraints.
- 'inference_scoring': the task core is automated through inference that produces a label, score, detection, classification, recognition, or forecast as the economically relevant output.
- 'informational_transformation': the task core is automated through generating, extracting, restructuring, translating, summarizing, or otherwise transforming informational content, where that transformed content is the economically relevant output.
- 'none': the task core is not meaningfully automated through the available channels.

Most important boundary:

- 'rule_based_workflow' = automation executes a predefined procedure
- 'planning_control' = automation selects, updates, or optimizes the procedure, allocation, route, schedule, or control action conditional on objectives and constraints

Channel-assignment checklist:

- first ask: what is the task-core output?
- then ask: through what mechanism does automation reach that output?
- if the output is a physical action on the world, start from 'physical_execution'
- if the output is correct execution of structured informational steps, start from 'rule_based_workflow'
- if the output is a selected or updated plan, route, schedule, allocation, or control choice, start from 'planning_control'
- if the output is an assessment, label, score, detection, recognition, or forecast, start from 'inference_scoring'
- if the output is transformed or generated informational content, start from 'informational_transformation'
- use 'none' only when no specific automation mechanism meaningfully reaches the task core
- if a specific mechanism is present but remains assistive only, 'exposure_level=1' may still take a non-none channel

Helpful contrasts:

- 'rule_based_workflow' versus 'planning_control':
 - workflow executes fixed steps
 - planning/control chooses or updates the steps, route, schedule, allocation, or control action
- 'inference_scoring' versus 'informational_transformation':
 - inference/scoring outputs an assessment, label, or forecast
 - informational transformation outputs content or transformed information
- 'planning_control' versus 'inference_scoring':

- if prediction feeds a schedule or routing choice, keep 'planning_control'
- if the prediction itself is the deliverable output, use 'inference_scoring'
- 'rule_based_workflow' versus 'informational_transformation':
 - if transformed text or extracted information only populates or routes a workflow, keep 'rule_based_workflow'
 - if the transformed text or extracted content is itself the output, use 'informational_transformation'

Boundary rules:

- Do not choose 'inference_scoring' merely because a predictive model exists upstream of planning, workflow, or physical execution.
- Do not choose 'informational_transformation' merely because an LLM is present as an interface, drafting aid, or explanation layer around another task-core output.
- If inference feeds planning, routing, scheduling, allocation, or control choice, the dominant channel remains 'planning_control'.
- If transformed information feeds deterministic step execution, the dominant channel remains 'rule_based_workflow'.
- If workflow execution uses inferred states or transformed content only to populate, prioritize, or route steps, the dominant channel remains 'rule_based_workflow'.
- If AI supports a physically automated task through vision, detection, or exception handling, the dominant channel remains 'physical_execution' unless the cognitive output itself has become the task core.

AI concept:

AI refers here to trained inferential systems that learn patterns from data and use those learned patterns, together with current inputs or context, to produce predictions, classifications, detections, transformed informational content, recommendations, or control-relevant outputs.

AI is not any advanced software in general.

These are not AI on their own:

- deterministic rules
- fixed workflow logic
- lookup tables
- hard-coded logic
- non-learned optimization
- ordinary software integration
- standard machinery without learned inference

Channel and AI are separate: assign the dominant channel first, then decide whether AI materially contributes. AI may be embedded in physical machinery, but that does not change the dominant channel unless the AI-generated output itself is the economically relevant task-core output.

AI integration:

- 'has_material_ai_integration=true' only if a learned inferential component is materially necessary to the economically relevant automation route for the task core.
- do not set 'has_material_ai_integration=true' merely because a learned component improves, pre-processes, or enriches an upstream input while the task would still be automated in essentially the same economically relevant way without it.
- ask: if the learned component were removed, would the task still be automated in essentially the same economically relevant way using deterministic rules, fixed logic, conventional software, or non-learned control?
- if yes, set 'has_material_ai_integration=false'
- do not infer 'has_material_ai_integration=true' from channel alone
- 'inference_scoring' can still be non-AI if it is deterministic scoring or fixed-rule classification
- 'informational_transformation' can still be non-AI if it is deterministic extraction, templating, or fixed transformation logic
- AI is a secondary layer, not a dominant channel
- do not let AI determine exposure level or substitution-versus-augmentation route by itself

Dominant AI function:

none | learned_state_inference | learned_content_transformation | learned_recommendation_decision | learned_adaptive_control

AI function definitions:

- 'learned_state_inference': the learned component mainly turns raw, noisy, or unstructured inputs into detected states, labels, classifications, scores, or forecasts.
- 'learned_content_transformation': the learned component mainly extracts, summarizes, translates, rewrites, restructures, or generates informational content.
- 'learned_recommendation_decision': the learned component mainly ranks, prioritizes, recommends, or proposes candidate choices used by a broader workflow or planning/control system.
- 'learned_adaptive_control': the learned component mainly maps sensed states or context into control-relevant actions, control parameters, or closed-loop policy updates in a physical or cyber-physical system.

AI function rules:

- choose exactly one dominant AI function when 'has_material_ai_integration=true'
- use 'none' when 'has_material_ai_integration=false'
- choose the function that best describes the learned-model contribution that is most central to the automation

route

- use 'learned_state_inference' when the learned output is mainly a detected state, label, classification, score, or forecast rather than a proposed action
- use 'learned_recommendation_decision' when the learned output mainly prioritizes, ranks, recommends, or proposes candidate actions or choices for a broader workflow or planning/control system
- if a machine uses learned perception to detect objects, states, or anomalies, keep 'dominant_channel=physical_execution' and set 'dominant_ai_function=learned_state_inference'
- if a machine uses learned closed-loop control or learned action policies, keep 'dominant_channel=physical_execution' and set 'dominant_ai_function=learned_adaptive_control'
- if learned perception only feeds workflow or planning, keep 'dominant_channel=rule_based_workflow' or 'planning_control' and use 'learned_state_inference' when AI materially contributes
- if learned ranking, prioritization, or candidate-choice generation feeds workflow or planning, keep 'dominant_channel=rule_based_workflow' or 'planning_control' and use 'learned_recommendation_decision' when AI materially contributes
- if learned content extraction, summarization, or rewriting only feeds workflow or planning, keep 'dominant_channel=rule_based_workflow' or 'planning_control' and use 'learned_content_transformation' when AI materially contributes
- when both learned perception and learned control are present, choose the AI function more central to the economically relevant automation route
- default to 'learned_adaptive_control' when learned control materially carries physical task execution
- default to 'learned_state_inference' when learned perception is the main learned contribution and control remains conventional

Explanation fields:

- 'short_rationale' must be one short sentence of at most 35 words
- it should explain the main economic reason for the assigned exposure and route in this country context, while remaining consistent with the dominant channel
- when possible, mention both the core economic reason and the country-context reason; if exposure_level <= 1, explain why the task remains assistive or non-automated rather than naming a predominant route
- 'substitution_route_summary' should be one short sentence describing the main plausible substitution route, or an empty string if no substitution path exists
- 'augmentation_route_summary' should be one short sentence describing the main plausible augmentation route, or an empty string if no augmentation path exists
- keep route summaries concrete and tied to the task core
- avoid generic references to "AI" or "software" without the mechanism by which the task core is affected

When uncertain:

- prefer the channel tied to the final economically relevant output rather than the most technically sophisticated ingredient
- do not overuse AI-related labels just because a modern system could include AI somewhere in the stack
- if two channels seem possible, choose the one that best describes what employers are economically buying when they automate this task in typical settings
- after choosing the mechanism, reassess exposure and route directly rather than inheriting them from the channel label

Consistency:

- if exposure_level = 0, then dominant_channel=none, has_material_ai_integration=false, dominant_ai_function=none, has_substitution_path=false, has_augmentation_path=false, margin=unclear, short_rationale="", substitution_route_summary="", augmentation_route_summary=""
- if exposure_level = 1, then has_substitution_path=false, has_augmentation_path=false, margin=unclear, substitution_route_summary="", augmentation_route_summary=""
- if exposure_level = 1, dominant_channel may be 'none' if no specific mechanism meaningfully reaches the task core, or a non-none assistive mechanism if one does
- if exposure_level = 1, has_material_ai_integration may be true or false depending on whether AI materially contributes to the assistive use
- level 1 still means assistive use only and not a credible labor-reallocation margin for a nontrivial share of the task core
- if dominant_channel=none, then exposure_level must be 0 or 1
- if dominant_channel=none, then has_material_ai_integration=false and dominant_ai_function=none
- if has_material_ai_integration=false, then dominant_ai_function=none
- if has_material_ai_integration=true, then dominant_ai_function must not be none
- if exposure_level >= 2, dominant_channel must not be none
- if has_substitution_path=false and has_augmentation_path=false, then margin=unclear
- if only one of the two path flags is true, margin will usually match that route
- if both path flags are true, margin may still be substitute, augment, both, or unclear depending on which route predominates

Output must be JSON only. No explanations outside the JSON.

User prompt template: Country-conditioned task automation classifier

```
O*NET TASK
SEMANTIC TITLE: {semantic_title}
COUNTRY CONTEXT: {country_name}
```

2.2 Income-Group Benchmark Prompt

System prompt: Income-group benchmark task automation classifier

You classify task-level economic automation exposure, mechanism, adjustment margin, and material AI integration for one standardized work task.

Input:

- short semantic title describing one O*NET task
- one World Bank income-group context

Scope:

- assess the task in a typical country in the named World Bank income group, using ordinary workplaces around 2026
- use 2026 as the fixed evaluation year for this prompt variant
- no frontier R&D assumptions
- no bespoke systems unless already common in typical countries in that income-group context

Typical income-group rule:

- "Typical" means deployable in ordinary workplaces in a typical country in the named World Bank income group around 2026, not frontier firms, not isolated pilots, and not bespoke donor- or vendor-supported implementations
- use the income-group context only as a coarse deployment benchmark
- do not infer a specific country, region, culture, or institution unless it follows directly from the task

Feasibility:

- feasible only if capability exists in commercial products or services, or widely used open-source
- deployment must be plausible in ordinary workplaces in a typical country in the named income-group context

Output:

Return only valid JSON with exactly these keys in this order:

- 1) dominant_channel
- 2) exposure_level
- 3) has_substitution_path
- 4) has_augmentation_path
- 5) margin
- 6) has_material_ai_integration
- 7) dominant_ai_function
- 8) short_rationale
- 9) substitution_route_summary
- 10) augmentation_route_summary

Decision order:

- first determine the task core as the essential output or time-dominant component
- then assign dominant_channel by the mechanism through which automation reaches the task core
- then determine economic exposure_level as a separate economic judgment
- then determine substitution and augmentation paths
- then determine the predominant margin
- then decide whether AI is materially integrated, and if so what the dominant AI function is
- then provide the short explanation fields consistent with those choices

Core concept:

This prompt measures economic automation exposure, not mere AI contact.

A task has economic automation exposure when currently available technology creates a credible labor-reallocation margin for a nontrivial share of the task core in the named income-group context: technology can perform that share at sufficient quality and reliability, and at low enough effective cost, that typical employers would reduce human labor input on that share.

Mere helpful or assistive contact that does not materially reduce human labor input on the task core does not count as economic exposure.

Exposure level:

0 = no credible economic automation margin

1 = assistive contact only; technology may help, but does not materially reduce human labor input on the task core

2 = meaningful partial economic exposure; a nontrivial share of the task core can be reassigned away from labor

, but humans remain central
3 = extensive economic exposure; a large share of the task core can be reassigned away from labor, or the task is mostly automatable in typical settings

Function-first rule:

- assign the dominant mechanism before judging substitution versus augmentation
- channel should describe the function through which automation reaches the task core
- 'exposure_level' and 'margin' are separate economic judgments and should not be inferred mechanically from the channel label alone
- do not lower or raise exposure merely because the mechanism belongs to a more familiar software route or a more visibly advanced technical route
- if the task core can be credibly reassigned away from labor for a nontrivial share of the task in the named income-group context, keep that exposure judgment even when the mechanism is ordinary software rather than frontier AI
- conversely, do not assign high exposure merely because the mechanism uses AI, optimization, or another advanced-seeming technology if cost, reliability, workflow fit, or income-group conditions do not support real labor reallocation

Margins:

substitute | augment | both | unclear

Margin concept:

Distinguish between:

- feasibility: whether there is a credible economically meaningful substitution path or augmentation path
- predominant margin: which route is more likely to characterize typical deployment for this task in the named income-group context around 2026

Margin rules:

- 'has_substitution_path=true' if there is at least one credible route by which technology could reduce the need for human labor on a nontrivial share of the task core
- 'has_augmentation_path=true' if there is at least one credible route by which technology could assist performance in a way that still reduces human labor input on a nontrivial share of the task core, even if a human remains central overall
- do not count tiny convenience gains, generic productivity boosts, or light-touch assistance as substitution or augmentation paths
- 'margin' should describe the predominant likely route in typical deployment in the named income-group context around 2026
- choose 'substitute' if substitution is the more likely and more economically important route for the task core
- choose 'augment' if augmentation is the more likely and more economically important route for the task core
- choose 'both' only if substitution and augmentation are both materially plausible and neither clearly dominates
- choose 'unclear' only if economic exposure exists but the available information is insufficient to judge the predominant route
- when one route is clearly more typical, more scalable, or more central to the task core, choose that route even if the other route is also possible

Route-independence rule:

- judge substitution versus augmentation after assigning channel, but as a separate economic object
- do not let channel choice determine the route unless the mechanism genuinely changes feasibility or the likely deployment pattern
- channel is a description of how automation happens, not evidence by itself that the route should become augmentation, substitution, or unclear

Dominant channel concept:

'dominant_channel' identifies the mechanism through which the task core is automated.

Choose the channel by the economically relevant output of the task core, not by upstream ingredients inside a larger system.

Each channel should read naturally as "the task core is automated through ..."

Physical versus cognitive is a consistency lens, not a separate output field.

Channel-assignment rule:

- assign channel before judging the predominant substitution-versus-augmentation route
- choose the channel that best describes the mechanism by which automation reaches the task core
- after assigning channel, judge exposure and route separately as economic objects
- do not infer low or high exposure from the channel label alone

Channels:

physical_execution | rule_based_workflow | planning_control |
inference_scoring | informational_transformation | none

Channel definitions:

- 'physical_execution': the task core is automated through machine-controlled physical actuation on objects,

- materials, equipment, or physical environments.
- 'rule_based_workflow': the task core is automated through deterministic execution of structured informational steps governed by explicit rules, predefined conditions, or fixed state transitions.
- 'planning_control': the task core is automated through computational selection or updating of plans, allocations, schedules, routes, or control actions under objectives and constraints.
- 'inference_scoring': the task core is automated through inference that produces a label, score, detection, classification, recognition, or forecast as the economically relevant output.
- 'informational_transformation': the task core is automated through generating, extracting, restructuring, translating, summarizing, or otherwise transforming informational content, where that transformed content is the economically relevant output.
- 'none': the task core is not meaningfully automated through the available channels.

Most important boundary:

- 'rule_based_workflow' = automation executes a predefined procedure
- 'planning_control' = automation selects, updates, or optimizes the procedure, allocation, route, schedule, or control action conditional on objectives and constraints

Channel-assignment checklist:

- first ask: what is the task-core output?
- then ask: through what mechanism does automation reach that output?
- if the output is a physical action on the world, start from 'physical_execution'
- if the output is correct execution of structured informational steps, start from 'rule_based_workflow'
- if the output is a selected or updated plan, route, schedule, allocation, or control choice, start from 'planning_control'
- if the output is an assessment, label, score, detection, recognition, or forecast, start from 'inference_scoring'
- if the output is transformed or generated informational content, start from 'informational_transformation'
- use 'none' only when no specific automation mechanism meaningfully reaches the task core
- if a specific mechanism is present but remains assistive only, 'exposure_level=1' may still take a non-none channel

Helpful contrasts:

- 'rule_based_workflow' versus 'planning_control':
 - workflow executes fixed steps
 - planning/control chooses or updates the steps, route, schedule, allocation, or control action
- 'inference_scoring' versus 'informational_transformation':
 - inference/scoring outputs an assessment, label, or forecast
 - informational transformation outputs content or transformed information
- 'planning_control' versus 'inference_scoring':
 - if prediction feeds a schedule or routing choice, keep 'planning_control'
 - if the prediction itself is the deliverable output, use 'inference_scoring'
- 'rule_based_workflow' versus 'informational_transformation':
 - if transformed text or extracted information only populates or routes a workflow, keep 'rule_based_workflow'
 - if the transformed text or extracted content is itself the output, use 'informational_transformation'

Boundary rules:

- Do not choose 'inference_scoring' merely because a predictive model exists upstream of planning, workflow, or physical execution.
- Do not choose 'informational_transformation' merely because an LLM is present as an interface, drafting aid, or explanation layer around another task-core output.
- If inference feeds planning, routing, scheduling, allocation, or control choice, the dominant channel remains 'planning_control'.
- If transformed information feeds deterministic step execution, the dominant channel remains 'rule_based_workflow'.
- If workflow execution uses inferred states or transformed content only to populate, prioritize, or route steps, the dominant channel remains 'rule_based_workflow'.
- If AI supports a physically automated task through vision, detection, or exception handling, the dominant channel remains 'physical_execution' unless the cognitive output itself has become the task core.

AI concept:

AI refers here to trained inferential systems that learn patterns from data and use those learned patterns, together with current inputs or context, to produce predictions, classifications, detections, transformed informational content, recommendations, or control-relevant outputs.

AI is not any advanced software in general.

These are not AI on their own:

- deterministic rules
- fixed workflow logic
- lookup tables
- hard-coded logic
- non-learned optimization
- ordinary software integration
- standard machinery without learned inference

Channel and AI are separate: assign the dominant channel first, then decide whether AI materially contributes. AI may be embedded in physical machinery, but that does not change the dominant channel unless the AI-generated output itself is the economically relevant task-core output.

AI integration:

- 'has_material_ai_integration=true' only if a learned inferential component is materially necessary to the economically relevant automation route for the task core.
- do not set 'has_material_ai_integration=true' merely because a learned component improves, pre-processes, or enriches an upstream input while the task would still be automated in essentially the same economically relevant way without it.
- ask: if the learned component were removed, would the task still be automated in essentially the same economically relevant way using deterministic rules, fixed logic, conventional software, or non-learned control?
- if yes, set 'has_material_ai_integration=false'
- do not infer 'has_material_ai_integration=true' from channel alone
- 'inference_scoring' can still be non-AI if it is deterministic scoring or fixed-rule classification
- 'informational_transformation' can still be non-AI if it is deterministic extraction, templating, or fixed transformation logic
- AI is a secondary layer, not a dominant channel
- do not let AI determine exposure level or substitution-versus-augmentation route by itself

Dominant AI function:

none | learned_state_inference | learned_content_transformation | learned_recommendation_decision | learned_adaptive_control

AI function definitions:

- 'learned_state_inference': the learned component mainly turns raw, noisy, or unstructured inputs into detected states, labels, classifications, scores, or forecasts.
- 'learned_content_transformation': the learned component mainly extracts, summarizes, translates, rewrites, restructures, or generates informational content.
- 'learned_recommendation_decision': the learned component mainly ranks, prioritizes, recommends, or proposes candidate choices used by a broader workflow or planning/control system.
- 'learned_adaptive_control': the learned component mainly maps sensed states or context into control-relevant actions, control parameters, or closed-loop policy updates in a physical or cyber-physical system.

AI function rules:

- choose exactly one dominant AI function when 'has_material_ai_integration=true'
- use 'none' when 'has_material_ai_integration=false'
- choose the function that best describes the learned-model contribution that is most central to the automation route
- use 'learned_state_inference' when the learned output is mainly a detected state, label, classification, score, or forecast rather than a proposed action
- use 'learned_recommendation_decision' when the learned output mainly prioritizes, ranks, recommends, or proposes candidate actions or choices for a broader workflow or planning/control system
- if a machine uses learned perception to detect objects, states, or anomalies, keep 'dominant_channel=physical_execution' and set 'dominant_ai_function=learned_state_inference'
- if a machine uses learned closed-loop control or learned action policies, keep 'dominant_channel=physical_execution' and set 'dominant_ai_function=learned_adaptive_control'
- if learned perception only feeds workflow or planning, keep 'dominant_channel=rule_based_workflow' or 'planning_control' and use 'learned_state_inference' when AI materially contributes
- if learned ranking, prioritization, or candidate-choice generation feeds workflow or planning, keep 'dominant_channel=rule_based_workflow' or 'planning_control' and use 'learned_recommendation_decision' when AI materially contributes
- if learned content extraction, summarization, or rewriting only feeds workflow or planning, keep 'dominant_channel=rule_based_workflow' or 'planning_control' and use 'learned_content_transformation' when AI materially contributes
- when both learned perception and learned control are present, choose the AI function more central to the economically relevant automation route
- default to 'learned_adaptive_control' when learned control materially carries physical task execution
- default to 'learned_state_inference' when learned perception is the main learned contribution and control remains conventional

Explanation fields:

- 'short_rationale' must be one short sentence of at most 35 words
- it should explain the main economic reason for the assigned exposure and route in the income-group context, while remaining consistent with the dominant channel
- when possible, mention both the core economic reason and the main income-group deployment reason; if exposure_level <= 1, explain why the task remains assistive or non-automated rather than naming a predominant route
- 'substitution_route_summary' should be one short sentence describing the main plausible substitution route, or an empty string if no substitution path exists
- 'augmentation_route_summary' should be one short sentence describing the main plausible augmentation route, or an empty string if no augmentation path exists

- keep route summaries concrete and tied to the task core
- avoid generic references to "AI" or "software" without the mechanism by which the task core is affected

When uncertain:

- prefer the channel tied to the final economically relevant output rather than the most technically sophisticated ingredient
- do not overuse AI-related labels just because a modern system could include AI somewhere in the stack
- if two channels seem possible, choose the one that best describes what employers are economically buying when they automate this task in typical countries in that income-group context
- after choosing the mechanism, reassess exposure and route directly rather than inheriting them from the channel label

Consistency:

- if exposure_level = 0, then dominant_channel=none, has_material_ai_integration=false, dominant_ai_function=none, has_substitution_path=false, has_augmentation_path=false, margin=unclear, short_rationale="", substitution_route_summary="", augmentation_route_summary=""
- if exposure_level = 1, then has_substitution_path=false, has_augmentation_path=false, margin=unclear, substitution_route_summary="", augmentation_route_summary=""
- if exposure_level = 1, dominant_channel may be 'none' if no specific mechanism meaningfully reaches the task core, or a non-none assistive mechanism if one does
- if exposure_level = 1, has_material_ai_integration may be true or false depending on whether AI materially contributes to the assistive use
- level 1 still means assistive use only and not a credible labor-reallocation margin for a nontrivial share of the task core
- if dominant_channel=none, then exposure_level must be 0 or 1
- if dominant_channel=none, then has_material_ai_integration=false and dominant_ai_function=none
- if has_material_ai_integration=false, then dominant_ai_function=none
- if has_material_ai_integration=true, then dominant_ai_function must not be none
- if exposure_level >= 2, dominant_channel must not be none
- if has_substitution_path=false and has_augmentation_path=false, then margin=unclear
- if only one of the two path flags is true, margin will usually match that route
- if both path flags are true, margin may still be substitute, augment, both, or unclear depending on which route predominates

Output must be JSON only. No explanations outside the JSON.

User prompt template: Income-group benchmark task automation classifier

```
O*NET TASK
SEMANTIC TITLE: {semantic_title}
INCOME GROUP CONTEXT: {income_group}
```

2.3 Context-Free Benchmark Prompt

System prompt: Context-free benchmark task automation classifier

You classify task-level economic automation exposure, mechanism, adjustment margin, and material AI integration for one standardized work task.

Input:

- short semantic title describing one O*NET task
- one generic ordinary-workplace benchmark context

Scope:

- assess the task in a generic ordinary workplace benchmark around 2026
- use 2026 as the fixed evaluation year for this prompt variant
- no frontier R&D assumptions
- no bespoke systems unless already common in ordinary workplace settings

Generic benchmark rule:

- "Generic ordinary workplace benchmark" means commercially plausible deployment under ordinary conditions that are neither frontier-only nor unusually resource-constrained
- this is not a country benchmark and not an income-group benchmark
- do not import country-specific assumptions about labor costs, infrastructure, regulation, or institutions

Feasibility:

- feasible only if capability exists in commercial products or services, or widely used open-source
- deployment must be plausible in ordinary workplaces under the generic benchmark context

Output:

Return only valid JSON with exactly these keys in this order:

- 1) dominant_channel
- 2) exposure_level
- 3) has_substitution_path
- 4) has_augmentation_path
- 5) margin
- 6) has_material_ai_integration
- 7) dominant_ai_function
- 8) short_rationale
- 9) substitution_route_summary
- 10) augmentation_route_summary

Decision order:

- first determine the task core as the essential output or time-dominant component
- then assign dominant_channel by the mechanism through which automation reaches the task core
- then determine economic exposure_level as a separate economic judgment
- then determine substitution and augmentation paths
- then determine the predominant margin
- then decide whether AI is materially integrated, and if so what the dominant AI function is
- then provide the short explanation fields consistent with those choices

Core concept:

This prompt measures economic automation exposure, not mere AI contact.

A task has economic automation exposure when currently available technology creates a credible labor-reallocation margin for a nontrivial share of the task core in the generic benchmark context: technology can perform that share at sufficient quality and reliability, and at low enough effective cost, that typical employers would reduce human labor input on that share.

Mere helpful or assistive contact that does not materially reduce human labor input on the task core does not count as economic exposure.

Exposure level:

0 = no credible economic automation margin

1 = assistive contact only; technology may help, but does not materially reduce human labor input on the task core

2 = meaningful partial economic exposure; a nontrivial share of the task core can be reassigned away from labor, but humans remain central

3 = extensive economic exposure; a large share of the task core can be reassigned away from labor, or the task is mostly automatable in typical settings

Function-first rule:

- assign the dominant mechanism before judging substitution versus augmentation
- channel should describe the function through which automation reaches the task core
- 'exposure_level' and 'margin' are separate economic judgments and should not be inferred mechanically from the channel label alone
- do not lower or raise exposure merely because the mechanism belongs to a more familiar software route or a more visibly advanced technical route
- if the task core can be credibly reassigned away from labor for a nontrivial share of the task in the generic benchmark context, keep that exposure judgment even when the mechanism is ordinary software rather than frontier AI
- conversely, do not assign high exposure merely because the mechanism uses AI, optimization, or another advanced-seeming technology if cost, reliability, workflow fit, or ordinary deployment conditions do not support real labor reallocation

Margins:

substitute | augment | both | unclear

Margin concept:

Distinguish between:

- feasibility: whether there is a credible economically meaningful substitution path or augmentation path
- predominant margin: which route is more likely to characterize typical deployment for this task in the generic benchmark context around 2026

Margin rules:

- 'has_substitution_path=true' if there is at least one credible route by which technology could reduce the need for human labor on a nontrivial share of the task core
- 'has_augmentation_path=true' if there is at least one credible route by which technology could assist performance in a way that still reduces human labor input on a nontrivial share of the task core, even if a human remains central overall
- do not count tiny convenience gains, generic productivity boosts, or light-touch assistance as substitution or augmentation paths
- 'margin' should describe the predominant likely route in typical deployment in the generic benchmark context around 2026
- choose 'substitute' if substitution is the more likely and more economically important route for the task

- core
- choose 'augment' if augmentation is the more likely and more economically important route for the task core
 - choose 'both' only if substitution and augmentation are both materially plausible and neither clearly dominates
 - choose 'unclear' only if economic exposure exists but the available information is insufficient to judge the predominant route
 - when one route is clearly more typical, more scalable, or more central to the task core, choose that route even if the other route is also possible

Route-independence rule:

- judge substitution versus augmentation after assigning channel, but as a separate economic object
- do not let channel choice determine the route unless the mechanism genuinely changes feasibility or the likely deployment pattern
- channel is a description of how automation happens, not evidence by itself that the route should become augmentation, substitution, or unclear

Dominant channel concept:

'dominant_channel' identifies the mechanism through which the task core is automated.

Choose the channel by the economically relevant output of the task core, not by upstream ingredients inside a larger system.

Each channel should read naturally as "the task core is automated through ..."

Physical versus cognitive is a consistency lens, not a separate output field.

Channel-assignment rule:

- assign channel before judging the predominant substitution-versus-augmentation route
- choose the channel that best describes the mechanism by which automation reaches the task core
- after assigning channel, judge exposure and route separately as economic objects
- do not infer low or high exposure from the channel label alone

Channels:

physical_execution | rule_based_workflow | planning_control |
inference_scoring | informational_transformation | none

Channel definitions:

- 'physical_execution': the task core is automated through machine-controlled physical actuation on objects, materials, equipment, or physical environments.
- 'rule_based_workflow': the task core is automated through deterministic execution of structured informational steps governed by explicit rules, predefined conditions, or fixed state transitions.
- 'planning_control': the task core is automated through computational selection or updating of plans, allocations, schedules, routes, or control actions under objectives and constraints.
- 'inference_scoring': the task core is automated through inference that produces a label, score, detection, classification, recognition, or forecast as the economically relevant output.
- 'informational_transformation': the task core is automated through generating, extracting, restructuring, translating, summarizing, or otherwise transforming informational content, where that transformed content is the economically relevant output.
- 'none': the task core is not meaningfully automated through the available channels.

Most important boundary:

- 'rule_based_workflow' = automation executes a predefined procedure
- 'planning_control' = automation selects, updates, or optimizes the procedure, allocation, route, schedule, or control action conditional on objectives and constraints

Channel-assignment checklist:

- first ask: what is the task-core output?
- then ask: through what mechanism does automation reach that output?
- if the output is a physical action on the world, start from 'physical_execution'
- if the output is correct execution of structured informational steps, start from 'rule_based_workflow'
- if the output is a selected or updated plan, route, schedule, allocation, or control choice, start from 'planning_control'
- if the output is an assessment, label, score, detection, recognition, or forecast, start from 'inference_scoring'
- if the output is transformed or generated informational content, start from 'informational_transformation'
- use 'none' only when no specific automation mechanism meaningfully reaches the task core
- if a specific mechanism is present but remains assistive only, 'exposure_level=1' may still take a non-none channel

Helpful contrasts:

- 'rule_based_workflow' versus 'planning_control':
 - workflow executes fixed steps
 - planning/control chooses or updates the steps, route, schedule, allocation, or control action
- 'inference_scoring' versus 'informational_transformation':
 - inference/scoring outputs an assessment, label, or forecast
 - informational transformation outputs content or transformed information

- 'planning_control' versus 'inference_scoring':
 - if prediction feeds a schedule or routing choice, keep 'planning_control'
 - if the prediction itself is the deliverable output, use 'inference_scoring'
- 'rule_based_workflow' versus 'informational_transformation':
 - if transformed text or extracted information only populates or routes a workflow, keep 'rule_based_workflow'
 - if the transformed text or extracted content is itself the output, use 'informational_transformation'

Boundary rules:

- Do not choose 'inference_scoring' merely because a predictive model exists upstream of planning, workflow, or physical execution.
- Do not choose 'informational_transformation' merely because an LLM is present as an interface, drafting aid, or explanation layer around another task-core output.
- If inference feeds planning, routing, scheduling, allocation, or control choice, the dominant channel remains 'planning_control'.
- If transformed information feeds deterministic step execution, the dominant channel remains 'rule_based_workflow'.
- If workflow execution uses inferred states or transformed content only to populate, prioritize, or route steps, the dominant channel remains 'rule_based_workflow'.
- If AI supports a physically automated task through vision, detection, or exception handling, the dominant channel remains 'physical_execution' unless the cognitive output itself has become the task core.

AI concept:

AI refers here to trained inferential systems that learn patterns from data and use those learned patterns, together with current inputs or context, to produce predictions, classifications, detections, transformed informational content, recommendations, or control-relevant outputs.

AI is not any advanced software in general.

These are not AI on their own:

- deterministic rules
- fixed workflow logic
- lookup tables
- hard-coded logic
- non-learned optimization
- ordinary software integration
- standard machinery without learned inference

Channel and AI are separate: assign the dominant channel first, then decide whether AI materially contributes.

AI may be embedded in physical machinery, but that does not change the dominant channel unless the AI-generated output itself is the economically relevant task-core output.

AI integration:

- 'has_material_ai_integration=true' only if a learned inferential component is materially necessary to the economically relevant automation route for the task core.
- do not set 'has_material_ai_integration=true' merely because a learned component improves, pre-processes, or enriches an upstream input while the task would still be automated in essentially the same economically relevant way without it.
- ask: if the learned component were removed, would the task still be automated in essentially the same economically relevant way using deterministic rules, fixed logic, conventional software, or non-learned control?
- if yes, set 'has_material_ai_integration=false'
- do not infer 'has_material_ai_integration=true' from channel alone
- 'inference_scoring' can still be non-AI if it is deterministic scoring or fixed-rule classification
- 'informational_transformation' can still be non-AI if it is deterministic extraction, templating, or fixed transformation logic
- AI is a secondary layer, not a dominant channel
- do not let AI determine exposure level or substitution-versus-augmentation route by itself

Dominant AI function:

none | learned_state_inference | learned_content_transformation | learned_recommendation_decision | learned_adaptive_control

AI function definitions:

- 'learned_state_inference': the learned component mainly turns raw, noisy, or unstructured inputs into detected states, labels, classifications, scores, or forecasts.
- 'learned_content_transformation': the learned component mainly extracts, summarizes, translates, rewrites, restructures, or generates informational content.
- 'learned_recommendation_decision': the learned component mainly ranks, prioritizes, recommends, or proposes candidate choices used by a broader workflow or planning/control system.
- 'learned_adaptive_control': the learned component mainly maps sensed states or context into control-relevant actions, control parameters, or closed-loop policy updates in a physical or cyber-physical system.

AI function rules:

- choose exactly one dominant AI function when 'has_material_ai_integration=true'
- use 'none' when 'has_material_ai_integration=false'

- choose the function that best describes the learned-model contribution that is most central to the automation route
- use 'learned_state_inference' when the learned output is mainly a detected state, label, classification, score, or forecast rather than a proposed action
- use 'learned_recommendation_decision' when the learned output mainly prioritizes, ranks, recommends, or proposes candidate actions or choices for a broader workflow or planning/control system
- if a machine uses learned perception to detect objects, states, or anomalies, keep 'dominant_channel=physical_execution' and set 'dominant_ai_function=learned_state_inference'
- if a machine uses learned closed-loop control or learned action policies, keep 'dominant_channel=physical_execution' and set 'dominant_ai_function=learned_adaptive_control'
- if learned perception only feeds workflow or planning, keep 'dominant_channel=rule_based_workflow' or 'planning_control' and use 'learned_state_inference' when AI materially contributes
- if learned ranking, prioritization, or candidate-choice generation feeds workflow or planning, keep 'dominant_channel=rule_based_workflow' or 'planning_control' and use 'learned_recommendation_decision' when AI materially contributes
- if learned content extraction, summarization, or rewriting only feeds workflow or planning, keep 'dominant_channel=rule_based_workflow' or 'planning_control' and use 'learned_content_transformation' when AI materially contributes
- when both learned perception and learned control are present, choose the AI function more central to the economically relevant automation route
- default to 'learned_adaptive_control' when learned control materially carries physical task execution
- default to 'learned_state_inference' when learned perception is the main learned contribution and control remains conventional

Explanation fields:

- 'short_rationale' must be one short sentence of at most 35 words
- it should explain the main economic reason for the assigned exposure and route in the generic benchmark context, while remaining consistent with the dominant channel
- when relevant, mention the core economic reason and the main generic deployment reason; if exposure_level <= 1, explain why the task remains assistive or non-automated rather than naming a predominant route
- 'substitution_route_summary' should be one short sentence describing the main plausible substitution route, or an empty string if no substitution path exists
- 'augmentation_route_summary' should be one short sentence describing the main plausible augmentation route, or an empty string if no augmentation path exists
- keep route summaries concrete and tied to the task core
- avoid generic references to "AI" or "software" without the mechanism by which the task core is affected

When uncertain:

- prefer the channel tied to the final economically relevant output rather than the most technically sophisticated ingredient
- do not overuse AI-related labels just because a modern system could include AI somewhere in the stack
- if two channels seem possible, choose the one that best describes what employers are economically buying when they automate this task in ordinary settings
- after choosing the mechanism, reassess exposure and route directly rather than inheriting them from the channel label

Consistency:

- if exposure_level = 0, then dominant_channel=none, has_material_ai_integration=false, dominant_ai_function=none, has_substitution_path=false, has_augmentation_path=false, margin=unclear, short_rationale="", substitution_route_summary="", augmentation_route_summary=""
- if exposure_level = 1, then has_substitution_path=false, has_augmentation_path=false, margin=unclear, substitution_route_summary="", augmentation_route_summary=""
- if exposure_level = 1, dominant_channel may be 'none' if no specific mechanism meaningfully reaches the task core, or a non-none assistive mechanism if one does
- if exposure_level = 1, has_material_ai_integration may be true or false depending on whether AI materially contributes to the assistive use
- level 1 still means assistive use only and not a credible labor-reallocation margin for a nontrivial share of the task core
- if dominant_channel=none, then exposure_level must be 0 or 1
- if dominant_channel=none, then has_material_ai_integration=false and dominant_ai_function=none
- if has_material_ai_integration=false, then dominant_ai_function=none
- if has_material_ai_integration=true, then dominant_ai_function must not be none
- if exposure_level >= 2, dominant_channel must not be none
- if has_substitution_path=false and has_augmentation_path=false, then margin=unclear
- if only one of the two path flags is true, margin will usually match that route
- if both path flags are true, margin may still be substitute, augment, both, or unclear depending on which route predominates

Output must be JSON only. No explanations outside the JSON.

User prompt template: Context-free benchmark task automation classifier

```
O*NET TASK
SEMANTIC TITLE: {semantic_title}
BENCHMARK CONTEXT: Generic ordinary workplace benchmark
```

3 Task-to-ISIC4 Pruning Voter

System prompt: Task-to-ISIC4 pruning voter

```
You are validating one candidate O*NET task linkage for one ISIC4 industry activity.

You will be given:
- one ISIC4 activity code
- one ISIC4 activity description
- one parent ISIC2 division code and label
- one candidate O*NET task semantic title
- one candidate O*NET task label
- one candidate O*NET task description
- one candidate occupation title

Decision rule:
Return 'is_valid=true' only if the candidate O*NET task is a materially plausible concrete task component of
the stated ISIC4 activity.

Guidelines:
- Precision is more important than recall.
- If unsure, return false.
- True if the task is a real work action that would plausibly occur inside firms carrying out that activity.
- False if the task is only loosely related to the product domain, supply chain, customer setting, or broad
  occupation family.
- False if the task is too generic to be meaningfully specific to the activity.
- False if the task belongs mainly to another industry workflow step or another production context.
- Multiple tasks may be valid for the same ISIC4 activity. Judge this edge independently.
- Focus on whether the task belongs inside the activity, not on how frequent or important it is.

Output valid JSON only with exactly:
{"is_valid": <boolean>}

No extra keys.
```

User prompt template: Task-to-ISIC4 pruning voter

```
ISIC4 ACTIVITY
CODE: {isic4_code_raw}
DESCRIPTION: {activity_description}
PARENT ISIC2: {isic2_from_isic4} {isic2_label}

CANDIDATE O*NET TASK
SEMANTIC TITLE: {onet_semantic_title}
LABEL: {task_label}
DESCRIPTION: {onet_task_desc}
OCCUPATION TITLE: {occupation_title}

Return JSON only.
```