# **NES Blueprint v2: Proposed Revisions and Enhancements**

## **Algorithmic Instantiation of Core Modules**

To strengthen the computational specificity of the NES, we detail each module’s inputs, processing (using pseudocode or equations), and outputs. This ensures the blueprint is implementable in algorithmic form , addressing any gaps in Blueprint v1.

### **Comparator Module – Drift-Diffusion Evidence Accumulator**

Inputs: The Comparator receives a set of competing impulses, each characterized by attributes: salience (stimulus-driven strength $S\_i$), normative congruence ($N\_i$), and urgency ($U\_i$) . These values may be initialized based on perceptual input (for $S\_i$), retrieved norms for the context (for $N\_i$), and time or context pressure (for $U\_i$). The module also takes thresholds ($\Theta\_{\text{upper}}, \Theta\_{\text{lower}}$) from the Assent Gate, defining decision bounds .

Processing: Each impulse $i$ accumulates a decision evidence variable $E\_i(t)$ over time via a multi-attribute drift-diffusion process . On each time-step $\Delta t$ (e.g. 10 ms), the Comparator updates evidence as:

* Drift rate calculation: $v\_i = w\_s S\_i + w\_n N\_i + w\_u U\_i + \mathcal{N}(0,\sigma)$ . Here $w\_s, w\_n, w\_u$ are weight parameters scaling the influence of salience, norm congruence, and urgency respectively, and $\mathcal{N}(0,\sigma)$ is Gaussian noise reflecting stochastic fluctuations. (From simulations, realistic ranges are $w\_s \approx 0.5$, $w\_n \approx 0.5$–$1.0$, $w\_u \approx 0.2$, and $\sigma \approx 0.1$–$0.3$ .) This equation integrates multiple factors: an impulse aligned with active norms (positive $N\_i$) gets an evidence boost, whereas one violating norms (negative $N\_i$) has its drift reduced or even reversed (toward a veto) . Urgency $U\_i$ can increase over time (modeling growing pressure to decide), e.g. $U\_i(t+\Delta t) = U\_i(t) + k\_u ,\Delta t$ , ensuring that indecision eventually resolves by nudging drift upward.
* Evidence integration: $E\_i \leftarrow E\_i + v\_i,\Delta t$. A small leakage or decay can be applied to unchosen impulses’ evidence (e.g. $E\_i \leftarrow E\_i \times 0.99$ per step) to prevent runaway integration or to model attentional fading .
* Threshold check: After each update, the Comparator checks if the evidence hits a bound: if $E\_i \ge \Theta\_{\text{upper}}$ (sufficient evidence for action), or $E\_i \le \Theta\_{\text{lower}}$ (sufficient evidence to veto) . If so, it outputs a tentative decision for impulse $i$ (“Yes/approve” or “No/veto”). Otherwise, no decision is output yet for $i$ this cycle.

Outputs: The Comparator’s primary output is a decision signal for each impulse that reaches threshold. It can output approval of an impulse (if $E\_i$ hits the upper bound) or rejection (if $E\_i$ hits the lower bound) . If multiple impulses are present and one crosses first, that impulse’s decision can be selected. The output includes the identity of the impulse and a flag indicating assent or veto. If no threshold is reached by a time limit (e.g. 3 s in simulations ), the Comparator defers to the RAA for further instructions (see RAA below).

Revision Note: The drift equation from v1 is retained , now annotated with typical parameter values from the Stroop simulations. For instance, setting a strong norm weight $w\_n=1.0$ yielded perfect accuracy (no threshold crossing for wrong impulses) but slower decisions, whereas $w\_n=0.5$ allowed some errors under noise. These ranges will guide implementers in choosing realistic parameters.

### **Assent Gate – Dynamic Threshold Modulation**

Inputs: The Assent Gate takes a baseline threshold $\Theta\_{\text{base}}$ (which could depend on context or task difficulty) and modulatory signals that adjust the threshold . Key inputs include a serotonin-like level (5HT) and the current emotion state, both of which influence caution vs. impulsivity. The Assent Gate can also receive meta-control instructions from the RAA (e.g. to tighten or relax control in certain situations) .

Processing: The Gate computes the effective decision bounds $\Theta\_{\text{upper}}$ and $\Theta\_{\text{lower}}$ for the current decision cycle as follows :

* Start from the baseline: $\Theta \leftarrow \Theta\_{\text{base}}$ (e.g. 1.0 in normalized evidence units ).
* Neuromodulator adjustment: Adjust $\Theta$ based on the serotonin level. A higher 5HT level (calm, inhibited state) raises the threshold, whereas lower 5HT (impulsive state) lowers it . For example, $\Delta \Theta\_{5HT} = k\_{\text{ser}} \times (\text{5HT}{\text{level}} - \text{normal}{5HT})$ . With $k\_{\text{ser}}>0$, if 5HT is above normal, $\Theta$ increases (requiring more evidence to act); if 5HT is below normal, $\Theta$ decreases . This models findings that high serotonin promotes waiting and behavioral inhibition, whereas low serotonin leads to impulsivity .
* Emotion adjustment: Modify $\Theta$ based on the affective state . “Hot” high-arousal emotions (anger, excitement) push the threshold downward (encouraging quick action), modeled as adding a negative term $\Delta \Theta\_{\text{hot}}$ . “Cold” moods (fearful caution, sadness) push the threshold upward (requiring more evidence), via a positive $\Delta \Theta\_{\text{cold}}$ . For instance, if anger is present, $\Theta \leftarrow \Theta + (-0.2)$ (making it easier to reach decision); if anxiety is present, $\Theta \leftarrow \Theta + 0.2$ (requiring more evidence to commit). These values can be tuned; Blueprint v1 suggested on the order of ±0.2 in normalized units as an example . The Affect subsystem provides the sign and magnitude of this adjustment each cycle.
* RAA directives: If the RAA (meta-controller) issues a command to modulate thresholds (e.g. in high-stakes contexts, raise $\Theta$ further ), the Gate incorporates that by an additional $\Delta \Theta\_{\text{RAA}}$.
* Bounds enforcement: Ensure the threshold stays within reasonable limits (e.g. $\Theta\_{\min}=0.1$ and $\Theta\_{\max}=2.0$ to avoid pathological values) .
* Set upper/lower bounds: Typically symmetric: $\Theta\_{\text{upper}} = \Theta$, $\Theta\_{\text{lower}} = -,\Theta$ (meaning equal magnitude for “yes” vs “no” evidence, though in principle an asymmetry could model bias toward action or inaction).

Outputs: The Assent Gate outputs the decision thresholds $\Theta\_{\text{upper}}, \Theta\_{\text{lower}}$ to the Comparator . These define how much evidence the Comparator needs to authorize or veto an action. In effect, the Gate gates actions by making it easier or harder for the Comparator to reach a decision criterion . No direct yes/no decision comes from the Gate itself; rather, it influences the Comparator’s likelihood of terminating.

Revision Note: We have made the Assent Gate’s algorithm explicit with pseudocode-style steps, incorporating the serotonin parameter $k\_{\text{ser}}$ and emotional adjustments . This formalization captures the mechanism behind NES’s “assent/withhold” function . The Stroop simulations confirmed the Gate’s expected behavior: lowering $\Theta$ (simulating low 5HT) dramatically sped responses , while raising $\Theta$ slowed responses , with accuracy trade-offs emerging only when norm weight was weakened.

### **Recursive Adjudication Algorithm (RAA) – Meta-Control Layer**

Inputs: The Recursive Adjudication Agent (RAA) monitors the global decision process. It receives: (a) Comparator state signals – e.g. whether evidence accumulation is stalemating or oscillating (indicative of conflict) , and if multiple impulses are present, their relative progress; (b) Norm Conflict Resolver outcomes – if invoked, whether a clear normative recommendation (“allow” or “veto”) was reached ; (c) Cycle count/time elapsed in the current decision – to detect protracted deliberation ; (d) Contextual stakes – the importance or urgency of a decision, which might be set externally or by internal appraisal (high stakes may demand more careful recursion) . In short, RAA observes both the quantitative state of deliberation (timing, evidence levels) and qualitative flags (conflicts, unresolved outcomes).

Processing: The RAA is a policy engine that can initiate additional decision-processing cycles or terminate deliberation based on meta-rules . Its algorithm can be outlined in pseudocode:

1. Conflict Detection: Check if the Comparator has failed to reach a decision within a normal time (e.g. no threshold hit after a typical response time) and if evidence $E\_i$ appears to hover without clear progress . Also check if Norm Conflict Resolver reported a tie or a close call. If so, label the situation as a conflict/impasse.
2. Initiate Recursion: In case of conflict, RAA triggers a recursive deliberation cycle . This may involve:  
   * Engaging the Norm Conflict Resolver (if not already done) to get a normative verdict on the dilemma . For example, if impulse X strongly violates norm Y, the Resolver might return “veto impulse” which RAA can then enforce.
   * Adjusting parameters: If conflict stems from nearly balanced factors, RAA can tweak the decision parameters for the next round. It might increase a norm weight to break a tie in favor of moral restraint, or escalate urgency $U\_i$ if a decision must be made swiftly . It can also temporarily raise or lower thresholds via the Assent Gate (e.g. raise $\Theta$ if a dangerous impulse is about to win, or lower $\Theta$ if indecision persists too long) .
   * Information seeking or re-evaluation: In a cognitive agent, RAA might prompt retrieval of additional information or reconsideration of assumptions. (In NES so far, this aspect is abstract, but one could imagine RAA querying memory for a guiding principle or considering a meta-norm if the conflict is severe .)
3. Arbitrate Multi-Impulse Priority: If multiple impulses compete, RAA decides how to handle them. It can focus processing on the most critical impulse first (suspending others), or run comparisons in parallel but sequentially address conflicts . For instance, if impulse A and B are both active, RAA might notice impulse A relates to a more urgent or norm-relevant situation and thus should be resolved before B .
4. Continue Iteration: RAA then signals the Comparator and Assent Gate to begin another cycle with the updated parameters or focus. Essentially, RAA acts as a loop controller: it allows the system to “think twice” or even multiple times about the decision . Each cycle can incorporate new adjustments or any learning that occurred.
5. Stopping Conditions: RAA monitors the number of recursive cycles and external time. It imposes a limit $N\_{\max}$ (e.g. perhaps 2–3 cycles for routine decisions, configurable based on stakes) on how many times it will iterate . If this limit is reached without a clear “Yes” decision, RAA will default to a “No/withhold action” decision . (Choosing inaction by default aligns with a conservative strategy: better to miss an opportunity than commit a serious violation in protracted uncertainty.) RAA may also stop recursion early if a cycle yields a decisive outcome (e.g. Norm Conflict Resolver issues an absolute veto, which RAA can immediately accept and terminate further deliberation ). Another stopping criterion: if urgency becomes extreme (the situation can’t wait), RAA will force whatever decision the current evidence favors (“leap to action”) or use a meta-norm like “when in stalemate and out of time, choose the safer option” .

Outputs: The RAA does not directly choose the object-level action; rather, it outputs meta-decisions and adjustments:

* It can output a command to stop deliberation along with an outcome (e.g. “veto impulse X now” if a norm veto condition was met, or “assent to X” if conflict resolved in favor of action).
* It outputs any parameter changes to apply in the next Comparator/Assent cycle (like updated thresholds or weights).
* If a decision is reached or forced, RAA outputs the final decision of the NES (which impulse, if any, is enacted) to motor/action systems.

Revision Note: The RAA’s logic is now explicitly delineated to dispel any hint of an infinite regress or homuncular “magic.” Every recursion step is rule-based and finite, governed by conditions like stalemate detection and cycle counts . For example, Blueprint v1 noted RAA stops after a set number of cycles and defaults to inaction if no resolution . Computationally, the cost of RAA’s meta-loop is modest: conflicts requiring recursion should be relatively rare or handled in a few cycles, and the Norm Conflict Resolver’s calculation is simple (a weighted sum and comparison) taking only milliseconds . Thus, RAA provides substantial flexibility (handling difficult decisions, persistent conflicts, multi-impulse scenarios) without unacceptable computational overhead.

### **Norm Repository – Structured Memory of Norms**

Inputs: The Norm Repository is queried with the current situation context and candidate impulse(s). Context can include environmental cues, task roles, or social setting (e.g. “library environment,” “emergency situation”), as well as the agent’s internal state (active goals, emotions) . The repository may also receive norm learning inputs: new norms from instruction or observation, feedback signals (rewards/punishments) for reinforcement learning of norms, and time-based triggers for decay.

Data Structure: Each Norm is stored with rich metadata :

* A value weight indicating its strength or priority (numeric, dynamic).
* An obligatoriness level (deontic tag): e.g. Strict Obligation, Prohibition, Guideline, Preference. Strict obligations/prohibitions may have a veto flag = True, meaning they can categorically block an impulse. Guidelines are weaker suggestions (veto flag = False, lower default weight).
* A context scope specifying when it applies (which situations, domains, or cue conditions).
* Source/origin information: how/when the norm was acquired (taught by authority, learned from culture, personal resolution). This can affect its malleability (e.g. norms learned from a revered authority might resist change).
* Last reinforcement timestamp or counter: when it was last confirmed or rewarded. This helps implement gradual decay for long-unused norms.

Additionally, we now categorize norms by type:

* Moral norms – core ethical rules (often internalized as absolutes). These typically carry high weight and often a veto flag (e.g. “never kill innocents” might be weight 0.9, veto=True). They change slowly if at all, and the system may set a floor on their weight (not decaying below a minimum).
* Social conventions – culturally or situationally dependent rules (etiquette, legal rules). These have moderate weight and are context-tagged (active only in relevant contexts). They can be overridden by moral norms if in conflict. They are more subject to change: if the agent’s environment changes (new culture or rules), convention norms can drift up or down as the agent observes different practices .
* Personal goals/prudential norms – the agent’s self-imposed principles or objectives (e.g. “stay healthy,” “save money”). These function as norms guiding behavior but are individualized. They may compete with moral or social norms (e.g. a personal goal to win might conflict with a moral norm against cheating). Their weights are often adjustable through the agent’s own reflections and experience.

Processing (Retrieval and Update): The Norm Repository operates in two modes:

* Retrieval mode: Given the current context and impulse(s), fetch all relevant norms. This means norms whose context scope overlaps with the current situation, or general norms with no specific context (always applicable moral norms). For each relevant norm, provide its weight and stance (oblige or forbid the impulse). The output is the set of norms (with weights, stances, veto flags) that apply to evaluating the impulse . This set is sent to the Comparator (for computing $N\_i$ contributions) and to the Norm Conflict Resolver if needed.
* Update mode: The repository supports functions to acquire or modify norms based on new information . For example:  
  + add\_norm(spec) to incorporate a new norm (e.g. after an authority figure explicitly communicates a rule, or the agent infers a norm from a scenario) .
  + update\_norm\_weight(norm\_id, Δ) to adjust a norm’s weight due to reinforcement or violation outcomes .
  + disable\_norm(norm\_id) or demote it, if it’s learned that the norm no longer applies (e.g. a law was repealed) .

Norm weights are dynamic. Each time a norm is successfully followed and yields expected positive outcomes (or the agent observes others following it with good results), that norm’s weight may incrementally increase – representing internalization or growing confidence in the norm. Conversely, if the norm is repeatedly irrelevant or following it leads to bad outcomes, its weight can drift downward . We implement a slow decay: over long periods of non-use or non-reinforcement, reduce weight slightly (tending toward a baseline) . If a norm is violated and no negative consequences occur, the agent might also weaken that norm (updating its belief that perhaps the norm is not strict). This models how norms “fade” if not maintained. However, certain moral norms might be set to not decay below a threshold without very strong contradictory evidence, reflecting their special status.

Conflict Emergence: Over time, as new norms are added or weights change, new conflicts can arise in the repository . For instance, acquiring a new value might clash with an older one. The repository should flag potential conflicts, especially between high-weight norms with opposing directives . This can prompt a one-time priority resolution process: the agent might set a general priority rule between those norms (effectively creating a meta-norm that Norm A > Norm B) . The repository can store these priority relationships (which the Conflict Resolver will use to break ties). This way, frequent conflicts lead to a learned resolution policy, reducing dithering if the same conflict occurs again.

Outputs: In retrieval mode, outputs the list of applicable norms with their parameters (weight, stance, veto, etc.) to be used by the Comparator and Conflict Resolver. In update mode, the output is an updated repository state (no direct output to other modules, but subsequent retrievals will reflect updates). The Norm Repository thus serves as both a knowledge base and a learning mechanism for the agent’s values , ensuring that NES’s normative guidance evolves realistically over time.

Revision Note: The Norm Repository is now more structured: we explicitly introduced norm types (moral, social, personal) and additional metadata tags, extending Blueprint v1’s tagging system. We have also clarified the dynamic norm update rules (internalization through reinforcement , decay through neglect , conflict-driven re-prioritization ). These additions make the Norm Repository an adaptive store rather than a static list – addressing how norms are acquired and change, which was identified as an area needing more specification.

### **Norm Conflict Resolver – Normative Arbitration Mechanism**

Inputs: The Norm Conflict Resolver is invoked when there is a direct conflict between norms regarding an impulse . It receives a set of two or more applicable norms that pull in opposite directions for a given decision. For a single impulse, this typically means at least one norm advocates for allowing the action and at least one norm advocates for prohibiting it . Each norm comes annotated with its weight (importance), its deontic stance (obligation/permission vs. prohibition), and any veto flag indicating an absolute rule . In more complex scenarios, conflicts could involve multiple impulses each supported by different norms; in such cases the Conflict Resolver might be asked to rank which impulse to favor (though usually RAA handles multi-impulse by focusing on one at a time).

Processing: The Conflict Resolver applies a weighted voting with veto algorithm :

1. Tally supportive vs. opposing weights: It sums the weights of norms favoring the impulse (call this $W\_{\text{yes}}$) and separately sums weights of norms opposing the impulse ($W\_{\text{no}}$) . For example, if NormA (weight 0.8) permits action and NormB (weight 0.5) forbids it, then $W\_{\text{yes}}=0.8$, $W\_{\text{no}}=0.5$.
2. Deontic veto check: Before comparing $W\_{\text{yes}}$ vs $W\_{\text{no}}$, the resolver checks if any prohibitive norm in the set carries an absolute veto flag . If a veto-flagged norm is present on the “no” side, it overrides the others . In the above example, if NormB had veto\_flag=True (“absolute prohibition”), the resolver would immediately decide to reject the impulse, regardless of NormA’s weight . (If there were a veto on the yes side – an absolute obligation – and a veto on the no side, that is an exceptional deadlock; NES would then require a higher-order resolution or default policy for such rare cases .)
3. Compare weighted sums: If no veto was invoked, the resolver compares $W\_{\text{yes}}$ and $W\_{\text{no}}$. If $W\_{\text{yes}} > W\_{\text{no}}$, the normative balance leans toward allowing the action; if $W\_{\text{no}} > W\_{\text{yes}}$, it leans toward forbidding the action. If they are roughly equal (within some indifference threshold), it registers a tie or stalemate.
4. Incorporate norm type hierarchy (if needed): In cases of close calls or ties, the resolver can apply secondary rules. For example, an obligation vs. obligation conflict might simply defer to weights, but a strict obligation might be given lexical priority over a guideline even if weights are similar . (Blueprint v1 noted that in practice the weight could encode this priority, but explicitly having a rule can help in design .)

Outputs: The Conflict Resolver outputs a decision recommendation: “Allow” the impulse, “Reject” the impulse, or “Tie/Unresolved” . It may also output a rationale (e.g. “Norm X vetoed” or “weights favored yes”) for logging or for RAA’s awareness. This output feeds into RAA and the Assent Gate/Comparator:

* If “Reject” (especially via veto), RAA/Assent Gate will enforce that by either directly terminating the impulse (veto signal) or by setting the Comparator’s evidence to a prohibitive state (e.g. a large negative $E$ or an effectively infinite threshold so it can’t cross) .
* If “Allow,” the RAA/Assent may conversely lower the threshold or boost that impulse’s drift to ensure it wins .
* If “Tie,” RAA knows further deliberation is needed. RAA might then either run another cycle hoping noise or urgency will break the tie, or seek additional input (e.g. ask for clarification, or apply a meta-norm rule like “choose the safer option in a tie”) .

Revision Note: The Conflict Resolver’s algorithm is now fully specified in stepwise form, aligning with the description in Blueprint v1 . Notably, this module implements the “voting and veto” decision policy that gives NES its normative decision character . This explicitness highlights a major innovation of NES: it is capable of computing a resolution when values clash, something absent in most other cognitive architectures . The weighted comparison combined with absolute constraints means NES can model scenarios like an inviolable moral rule vetoing a tempting action, or two competing principles where one slightly outweighs the other after deliberation. These fine-grained outcomes (including the possibility of indeterminacy) are crucial for modeling moral dilemmas and complex executive decisions.

## **Refined Norm Repository Dynamics and Structure**

To be inserted as an expanded description in the Norm Repository section (Blueprint v1 Section II.C):

Norm Types and Parameters: We categorize internalized norms into (1) moral absolutes, (2) social conventions, and (3) personal goals/preferences, as noted above. This categorization affects their parameters:

* Moral absolutes typically carry highest priority. They often come with veto\_flag=True (for prohibitions) or an imperative to always act (for obligations) because they represent non-negotiable values. Their weight is usually high (e.g. 0.8–1.0 on a 0–1 scale) and maintained with minimal decay. The system might enforce a floor weight for certain sacred values, ensuring they remain influential unless explicitly relearned.
* Social conventions have intermediate status. They guide behavior in particular contexts (e.g. politeness, legal norms) and thus carry context tags. Their weight can range moderate (e.g. 0.3–0.7) and is updated based on observed social reinforcement. They are overrideable by moral norms (if conflict, a strong moral norm will usually win) and more prone to drift if the agent’s environment changes .
* Personal goals/preferences function as norms the agent chose or learned for itself (like “exercise daily,” “maximize profit”). They might be encoded as norms with a certain weight and positive stance for relevant actions. These often lack veto power and can be outweighed by moral or social norms in conflicts. They are quite malleable: the agent can revise its own goals through reflection or experience. In effect, personal norms link the planning/utilitarian aspect of decision-making into the normative framework (the agent’s long-term goals become part of what NES considers).

By differentiating norms in this way, NES v2 clarifies that not all norms are equal – some are inviolable, some situational, some self-chosen – and the system’s behavior adjusts accordingly. For example, a strict moral norm (“do no harm”) might override a personal goal (“win the competition”) without hesitation, whereas two social norms might simply be weighed by context relevance.

Norm Acquisition: NES acquires norms through multiple pathways:

* Instruction and Imitation: The agent can internalize norms given by authority or by imitating others. Blueprint v1 described a Bayesian learning mechanism : the agent updates its belief in a candidate norm when observing social outcomes (e.g. seeing someone punished suggests a norm against that action). This is implemented by adjusting a norm’s weight upward when evidence for it is observed, or adding a new norm entry when a threshold of evidence is reached. For instance, if entering a library and observing admonishments for loud behavior, the agent infers a norm “be quiet in library” and adds it with some initial weight .
* Reinforcement Learning: The agent’s own actions and outcomes strengthen or weaken norms. Each time the agent follows a norm and gets a positive outcome (reward or no negative consequence), the norm’s weight is increased slightly . If the agent violates a norm and encounters punishment or guilt (an internal negative reinforcement), that norm’s weight will increase (deterring future violation) . Conversely, if the agent violates a norm and nothing bad happens (or it yields a reward), the norm’s weight may decrease – though moral norms might require repeated disconfirmations to significantly drop. This mechanism ensures the Norm Repository self-tunes to the agent’s lived experience, analogous to how humans learn the true importance of rules.
* Explicit Reappraisal: The agent can deliberately modify its norms via reflection or meta-cognitive evaluation. NES can support a process whereby the agent questions a norm (“Is this rule valid for me?”) and potentially changes its stance. For example, after joining a new culture, an agent might decide an old norm is no longer applicable and consciously down-regulate it. This could be triggered by persistent conflict: if Norm A and Norm B conflict often, the agent might decide to generally prioritize one (a meta-norm update as discussed) . Such reflective updates are less automatic but are important for long-term norm adaptation (they could be handled by the RAA invoking a special learning routine outside of immediate decision cycles).

Norm Drift and Decay: To prevent outdated norms from unduly influencing decisions, NES implements slow drift in norm weights:

* Norms that are rarely used or reinforced will gradually decay in strength . A small decrement (e.g. a few percent) could be applied each week of inaction, for instance. This models the forgetting or deemphasizing of norms that are no longer relevant in the agent’s life (e.g. rules from childhood that fade in adulthood if not reinforced).
* Norms that are frequently confirmed may slowly strengthen (the opposite of decay), plateauing as they become core to the agent’s identity.
* The decay mechanism uses the “last reinforced time” tag: if a long period has passed since reinforcement, a tiny downward adjustment is applied, ensuring norms require sustained support to remain strong .
* Crucially, decay will not reduce a norm’s weight below a minimum if it has a protected status (for example, moral absolutes might never drop below weight 0.5 even if not recently reinforced). This prevents losing critical ethical constraints merely due to lack of recent activation.

Conflict Emergence and Resolution: The repository now proactively tracks norm conflicts. If it detects that two high-weight norms are frequently applicable together but give conflicting prescriptions (e.g. “be honest” vs “be kind” in situations where truth may hurt), it flags this to the meta-level. NES can then establish a priority ordering or meta-norm to handle this conflict in general . For instance, the agent may adopt a meta-norm like “Honesty outweighs sparing feelings when they conflict.” This is stored in the repository (perhaps as a rule that in context XYZ, Norm A > Norm B). The next time both norms fire, the Conflict Resolver can immediately apply this hierarchy , speeding up the decision. Essentially, the Norm Repository not only stores norms but also evolving knowledge about relationships between norms.

These refinements ensure the Norm Repository in Blueprint v2 is comprehensive and dynamic. It serves not just as static memory, but as a living system of values – growing with new experiences, gradually adapting to change, and organizing itself to reduce internal conflict. This addresses prior gaps by clearly distinguishing types of norms and explaining how norms are acquired and tuned over time, which is a point that sets NES apart from other models that often treat “goals” or “values” more abstractly .

## **RAA Recursion Refinement and Termination Logic**

To be integrated into the RAA module description (Blueprint v1 Section II.E), expanding on recursion management:

We clarify here the triggers, limits, and rationale for the Recursive Adjudication Algorithm’s iterative cycles:

* Conflict Triggers: RAA monitors for signs of conflict or indecision. A key trigger is a prolonged decision time without threshold crossing . In human terms, if a decision that should be made in, say, under 1 second is still unresolved after several seconds of internal processing, it indicates the evidence is evenly balanced or oscillating. Another trigger is a direct signal from the Norm Conflict Resolver: if it returns a “tie” or if it vetoes an impulse that the Comparator was about to approve (or vice versa), that discrepancy prompts further reflection . Finally, if multiple impulses are active and competing (e.g., two actions both have moderate evidence), RAA treats this as a complex conflict scenario requiring arbitration . These conditions are detected via the inputs described (e.g. conflict flags, multiple active impulses, norm stalemates).
* Recursion Depth and Resource Use: NES v2 explicitly limits recursion to a reasonable depth $N\_{\max}$ to avoid infinite loops . This can be a fixed small number (like 3 cycles) or adaptive based on context (e.g., allow more cycles if the stakes are life-and-death, fewer if time-critical). Each cycle uses additional time and cognitive resources, so $N\_{\max}$ balances better decision-making with timeliness and computational cost. By design, typical decisions should resolve in 1 cycle; only thorny dilemmas invoke RAA recursion. Empirically, we expect perhaps 0–2 extra cycles in most cases, which is akin to a person “thinking twice or thrice” at most. This ensures tractability – NES does not get stuck in analysis paralysis.
* Stopping and Termination: We reiterate and formalize the stopping criteria to ensure no unbounded regress . RAA will terminate deliberation when:  
  1. A cycle yields a clear outcome (one impulse approved or all impulses vetoed) – then recursion is unnecessary further.
  2. The maximum cycle count $N\_{\max}$ is reached without a decision – then RAA forces a resolution. The default policy is withhold action if no go-decision has satisfied all checks . This default to inaction is justified by NES’s normative focus: if after exhaustive checking the system isn’t confident an action is norm-compliant or advantageous, refraining is safer. However, RAA could implement alternative defaults if appropriate (e.g., in a critical survival context, doing something might be better than inaction – such meta-decision rules could be context-dependent).
  3. An external interrupt or higher-level system aborts (if NES were part of a larger architecture that imposes an upper time bound, for instance).
* The termination output includes a marker of how the decision was reached (e.g., “decided after 2 recursive cycles” or “timed-out, defaulted to no-go”).
* Computational Considerations: Each additional cycle involves re-running the evidence accumulation (possibly under new parameters) and/or conflict resolution. The design assumes these processes are fast; indeed, our Stroop-task simulation runs hundreds of trials with 0.01 s time steps comfortably because each step is a simple drift update . The Norm Conflict Resolver’s computation is a small number of arithmetic operations on a handful of norms . Thus, even a few extra cycles do not significantly burden a cognitive agent or a neural process – they might correspond to a few hundred milliseconds more of brain activity. This addresses concerns that recursion might be intractable: in practice, it’s a bounded, efficient re-check that adds robustness. Moreover, the urgency signal ensures that if cycles pile up, the drift rates increase to push the system toward a conclusion , preventing endless dithering.
* No Hidden Homunculus: Importantly, the RAA itself is not a mysterious decider outside the system – it follows preset rules (as above) for when to loop or stop. There is no need for an “inner executive” to subjectively decide “OK that’s enough thinking”; the blueprint encodes that logic explicitly (e.g. a counter for cycles, a threshold for urgency) . For example, we might implement RAA as a simple loop in code with a for cycle in range(N\_max): ... break if decision\_made. The moment RAA hits the loop limit, it invokes the default action. This transparent criterion means the model avoids the infinite regress of a homunculus deciding when the homunculus should stop – the stopping is triggered by a mechanistic rule.

By refining the RAA in this way, NES v2 makes its meta-control layer as concrete as the object-level decision process. We foresee testing these recursion limits in simulation: e.g., induce a difficult conflict and verify that the model indeed stops after $N\_{\max}$ loops and defaults to a veto (and examine if that aligns with what a person might do under extreme indecision). Overall, RAA’s refinements shore up NES’s claim to have a self-reflective yet finite executive process – capturing the intuition of “I tried to decide, couldn’t with certainty, so I held back,” in a reproducible algorithm.

## **Enhanced Emotion-Cognition Integration (Rapid Affective Appraisal)**

In Blueprint v1, emotions modulated the Assent Gate’s threshold . Blueprint v2 expands the role of emotions via a Rapid Affective Appraisal mechanism that influences multiple decision parameters in real time:

* “Hot” vs “Cold” Distinction: We clearly distinguish hot emotional states (high arousal, intense affect) from cold states (low arousal, subdued or neutral affect) and detail their effects on decision dynamics . In a hot state (e.g. anger, excitement, fear), the agent’s immediate responses are biased towards quick action – implemented by lowering decision thresholds and potentially biasing drift rates upward. For instance, anger not only lowers $\Theta$ (less evidence needed to act) , but can also inject a positive urgency or automatic bias toward acting (increasing $U\_i$ temporarily for impulses congruent with that anger). Conversely, cold states (calm deliberation, sadness, fatigue) raise thresholds (more evidence needed) and could even dampen drift (e.g. reducing $U\_i$ or adding a small negative drift component reflecting caution). These dual effects mean emotion influences both when a decision threshold is reached and how quickly evidence accumulates.
* Norm Salience under Emotion: Emotions don’t treat all norms equally; they make some norms more salient and others less. NES v2 encodes this by allowing the Norm Repository’s weights to be momentarily adjusted by emotional context . For example, in a state of intense anger, norms about not causing harm might be down-weighted slightly, while norms about self-assertion or justice are up-weighted (anger often prioritizes perceived fairness or retaliation). Blueprint v1 gave an example: anger could decrease weight of “do no harm” and increase weight of “stand up for yourself” . We formalize this: each norm can have an associated emotional tag indicating if a particular emotion tends to amplify or suppress it. When that emotion is active, the norm’s effective weight $w\_j$ is multiplied by a factor (e.g. $1 + \lambda$ or $1 - \lambda$) to represent this bias . Likewise, empathy or love (a positive emotion) might increase weights of pro-social norms (“help others”) , whereas fear might amplify norms of safety but suppress bold action norms. These weight shifts feed into the Comparator’s $N\_i$ calculation, directly altering drift rates in line with the emotional priority.
* Emotion and Working Memory: A hot emotional event can cause certain information to persist longer in the deliberation. NES v2 accounts for this by letting emotions affect the decay of evidence and the persistence of impulses in RAA cycles. If an impulse has a strong emotional tag (say it triggered anger or fear), RAA will allow it to carry over across cycles more strongly . Technically, we reduce the decay factor for that impulse’s evidence or mark it as important so it isn’t dropped from consideration. Blueprint v1 suggested a formula where the evidence decay rate is modulated by emotional salience – high emotional salience yields a decay factor closer to 1 (meaning little forgetting) . For example, an insult (anger-triggering) will not be easily dismissed; the impulse to retaliate stays active through recursive deliberation, which corresponds to our experience of “not being able to let something go” when emotional . Cold emotions or neutral states do the opposite: they let unimportant impulses fade normally or even faster (e.g. apathy might cause even normally significant considerations to drop off) .
* Attention Bias: Emotions skew what the agent focuses on. In NES terms, this can be modeled by how the Comparator selects or emphasizes impulses. If the agent is afraid, any impulse related to addressing the threat might receive a higher initial salience $S\_i$ (or an extra boost to its evidence) compared to other impulses . Fear effectively shines a spotlight on threat-relevant choices (the norm “ensure safety” might jump to top priority ). A happy or curious mood might broaden consideration, allowing more impulses to be processed in parallel but each less forcefully. We integrate this by allowing the Affect system to influence the distribution of initial activation among impulses and even which impulses enter the Comparator first.
* Temporal Decay of Emotional Effects: Crucially, emotional influences are not static – a rapid affective appraisal has a fast onset and then decays. NES v2 introduces explicit affective decay curves. For modeling, we assume emotions follow a roughly exponential decay unless sustained: $\text{EmotionIntensity}(t) = E\_0 , e^{-t/\tau}$, where $\tau$ is the decay constant. A hot emotion has a relatively short $\tau$ (it spikes and decays over seconds or minutes if not re-triggered), whereas a cold mood might have a longer, low-level influence. We apply this to threshold and norm adjustments: e.g., if anger flares at $t=0$ with an effect of $\Delta\Theta=-0.3$, after some time (depending on $\tau\_{\text{anger}}$) that effect will shrink back toward 0. This means if an impulse can be stalled, the hot urgency might literally cool off. The RAA could exploit this: in some cases, delaying a decision might be a strategy to allow a hot emotion to subside (analogous to “count to ten when angry”). Our architecture supports that, since each new cycle RAA can recalc thresholds with updated (likely lower) emotion inputs if the emotion is decaying. Likewise, norm weight biases from emotion will fade – the Norm Repository could gradually restore weights to baseline after the emotional spike passes. For implementation, we maintain an emotional state vector with current levels for each relevant emotion which update with decay each timestep unless renewed.

In summary, the Rapid Affective Appraisal enhancements give NES a richer emotion-cognition loop. Emotions now dynamically tune norm evaluation , evidence accumulation, threshold setting , and the persistence of thoughts . This ensures that NES can capture phenomena like:

* Why we might make a rash choice in the heat of the moment (low threshold, norm biases favoring action), but if we hesitate a bit, cooler reasoning can prevail (thresholds reset higher as emotion decays).
* How emotional salience can cause certain decisions to take longer because the system is essentially too focused on an emotionally charged aspect, potentially requiring extra cycles (e.g. ruminating over an insult).
* How an executive system can integrate the rapid “good/bad” gut reactions (System 1, in dual-process terms) with the slower normative reasoning (System 2) . NES does so by allowing the gut reaction to immediately influence the parameters of deliberation, but not to entirely bypass deliberation.

These additions address the request to deepen the affective component of NES. By specifying how “hot” and “cold” states quantitatively alter decision parameters and by giving them time courses, we make testable predictions (e.g., Section “Experimental Predictions” below outlines how varying an emotional parameter should change outcomes). NES thus becomes a model of an affect-informed executive, integrating emotion and rational norm-based control in a unified architecture .

## **Comparative Distinctions from Other Frameworks**

NES v2 explicitly highlights how it converges with or diverges from several major cognitive frameworks, addressing points where clearer contrast was needed. Below we summarize key distinctions:

* Global Workspace Theory (GWT): GWT proposes a central “workspace” where information is broadcast to the system, but it does not define a specific decision-maker – control emerges from whichever information wins access to the workspace. GWT effectively has an implicit gating via attention/competition, not a dedicated assent module . NES vs GWT: NES provides a concrete gating mechanism (Assent Gate) that decides which impulse is allowed to drive action , whereas GWT would leave such inhibition to whatever unconscious processes might be configured. Additionally, NES’s Comparator+RAA actively adjudicates inputs against norms, while GWT is content-agnostic (norms could influence the competition in GWT, but GWT doesn’t specify how) . NES can be seen as a specific instantiation that could function within a global workspace – e.g., once NES approves an action, that decision could be the content that enters the global workspace (the conscious thought “I will do X”). But unlike GWT, NES builds in a value-based filter before global broadcasting. In short, both have a notion of a bottleneck (one decision at a time), but NES defines the bottleneck’s decision policy explicitly.
* Predictive Coding (PC) and Free Energy: Predictive coding describes the brain as a hierarchy minimizing prediction errors, with no centralized executive. Control is distributed across levels of prediction; “decisions” emerge from whichever prediction suppresses error. NES vs Predictive Coding: NES introduces an explicit executive layer (RAA and conflict resolver) that computes what to do when competing predictions (or impulses) conflict, whereas pure PC would just let the “error dynamics” play out. Importantly, PC doesn’t have a notion of normative evaluation – it’s driven by statistical/goal expectations. NES adds a normative dimension: rather than solely minimizing prediction error or surprise, it checks actions against internalized ideals. One can think of NES as adding a deontic constraint to a predictive mind: even if an action would reduce prediction error (e.g. fulfilling a habit), NES might veto it because it violates a norm. Mechanistically, one could map NES’s comparator to a form of precision-weighted prediction error accumulation (with norms modulating precision of certain prediction errors), but NES stands out by having a top-down rule-based veto, which predictive coding frameworks lack in their basic form .
* Hierarchical Reinforcement Learning (HRL): HRL uses a hierarchy of policies (meta-controller, sub-controllers) to tackle tasks at multiple timescales. It’s goal-driven and each layer optimizes reward, delegating to lower layers. NES vs HRL: Both are hierarchical control systems, but the content of control differs. In HRL, the top layer’s “goal” is just a more abstract reward-driven objective; in NES, the top layer’s mandate is adherence to norms (which might not align with immediate reward) . NES’s RAA+Norm Resolver acts akin to an HRL meta-policy that can veto lower policies, except its criterion is normative utility rather than cumulative reward . HRL’s hierarchy is trained or configured to maximize returns, whereas NES’s hierarchy is partly hand-coded with ethical priorities (though it can learn norms, it treats them as constraints) . Another difference: HRL inherently distributes the executive role (each layer making decisions for its scope) , which is similar in NES (comparator handles immediate evoked impulses, RAA handles meta-decisions). However, HRL typically doesn’t have an explicit veto concept – a high-level policy might override a low-level action if it’s suboptimal, but that’s framed in terms of expected reward. NES formalizes veto for normative reasons even if an action is locally appealing. Both approaches avoid a single monolithic homunculus by decomposing decisions, but NES is more prescriptive about layers (it has specific modules for evidence accumulation vs norm arbitration), whereas HRL’s layers are emergent or designed for task subgoals . In essence, NES is to moral/meta control what HRL is to goal/task control – they parallel structure, yet NES could enforce something like Asimov’s laws (hard constraints) which HRL would have to encode as part of the reward function implicitly .
* ACT-R (and SOAR) Cognitive Architectures: ACT-R posits a central production system that selects one rule (production) at a time to fire, integrating different modules (memory, vision, etc.). It has a sequential decision loop with conflict resolution based on rule utility. NES vs ACT-R: NES differs by using a continuous, parallel evidence accumulation for decisions rather than a discrete rule conflict resolver. In ACT-R, if two rules conflict, a conflict resolution scheme (like utilities or conflict set) picks one – akin to a single-step winner. NES’s Comparator instead integrates evidence over time, which can capture more nuanced conflicts (e.g. a strong norm slowly overriding a strong impulse). Also, ACT-R does not inherently have a normative module; any norm has to be encoded as a production rule condition or a goal. NES provides a dedicated Norm Repository and Conflict Resolver that operate on principles beyond utility – including absolute prohibitions , which ACT-R would have to emulate via carefully crafted production priorities. NES’s executive is more distributed (accumulators + gating + recursive check) compared to ACT-R’s centralized production selection loop . However, NES aligns with ACT-R in being implementable – both specify clear operations and could be turned into a program . The key governance difference is NES’s emergent decision from parallel evaluations vs. ACT-R’s single-threaded rule engine . One could imagine implementing NES within an ACT-R-like system by having the comparator and assent gate as a set of productions continuously updating a decision variable, but it would be an extension of the ACT-R paradigm to include drift dynamics and normative weighting explicitly.
* Dual-Process (System 1 vs System 2) Theories: These psychological theories distinguish fast, automatic, emotional responses (System 1) from slow, deliberative, rational thought (System 2). They usually don’t provide an algorithm, just the idea that an override of impulse by deliberation can occur. NES vs Dual-Process: NES can be seen as providing the mechanistic bridge between System 1 and System 2 . The Comparator (especially with emotion effects) models the intuitive, automatic accumulation of affordances and impulses (including a “gut” norm signal $N\_i$ for moral intuitions), while the RAA and Conflict Resolver implement more reflective evaluation (similar to a controlled process deciding whether to allow the impulse). Unlike generic dual-process models, NES doesn’t leave the interaction mysterious: it has a clear point where the slower process (normative check) can veto the faster one (“assent gate” acting as System 2 veto or guidance over System 1 impulses) . Dual-process theory would predict, for example, that under cognitive load (i.e. System 2 weakened), people make more impulsive choices – NES predicts the same: if the Assent Gate threshold is lowered or RAA recursion curtailed (simulating reduced executive capacity), the impulsive drifts will more often trigger action. A difference is that dual-process typically frames moral judgment as either a quick intuition or a reasoned judgment; NES shows how they integrate every time an action decision is made – the “intuitive” inputs (salience, initial norm signals possibly from emotion) continuously feed the Comparator, while the reflective layer actively checks and balances. So NES operationalizes System 2 override as a combination of a raised threshold plus possible multiple cycles of evidence checking, which is a novel computational interpretation of dual-process ideas .
* Higher-Order Theories of Consciousness (HOT): HOT claims a mental state is conscious only if one has a higher-order representation of it. While NES is not a theory of consciousness per se, it intersects in that it posits a meta-level (RAA) monitoring first-order processes. However, NES’s focus is on decision control, not on explaining subjective awareness. Where HOT might ask “when do I become aware of an urge?”, NES asks “when do I allow an urge to turn into action?”. Thus, NES avoids directly addressing HOT’s domain. That said, NES’s structure is consistent with having a higher-order monitoring of impulses (RAA observing comparator states could be seen as a kind of higher-order representation of the decision process). But NES does not require that representation to be conscious – it can be entirely sub-personal. In fact, NES could complement HOT by suggesting that what we call an act of will (a conscious decision) is the moment the RAA/Assent Gate commits to an action and that content is broadcast (which could then become a higher-order thought).
* Executive Function Theories (e.g. Supervisory Attentional System (SAS), Conflict Monitoring model): Psychological models like Norman & Shallice’s SAS proposed a top-level supervisor that inhibits inappropriate schemas , and the conflict monitoring model (Botvinick, Cohen, et al.) suggests ACC detects conflict and signals DLPFC to increase control. NES vs SAS/Conflict Monitoring: NES is very much in line with these, but more explicit. The SAS was essentially a homunculus placeholder – NES provides the details SAS lacked (how does the supervisor decide? via drift diffusion and norm checks, in our model) . The conflict monitoring theory is reflected in NES’s Comparator generating conflict signals (low drift or oscillation when impulses and norms clash) and in RAA’s ability to adjust control (like raising thresholds when conflict is high, analogous to ACC->PFC signaling) . NES can be seen as an instance of a Supervisory Attention-like system, with the Norm Conflict Resolver and RAA fulfilling the role of a supervisory controller that kicks in when routine processing (impulse-driven) isn’t resolving cleanly . The major addition NES brings is the normative dimension – SAS and related models focus on goal-driven conflict (like task rules vs impulses), whereas NES explicitly includes moral/social norms as first-class elements in conflict detection and resolution . This makes NES’s “supervisor” concerned not just with task performance, but with rule compliance, a leap that standard executive function models typically didn’t formalize (they assumed any such “values” are just part of the goal structure).

In sum, NES’s unique contributions relative to existing frameworks are:

* Normative gating: A dedicated mechanism for internalized norms to directly permit or block actions , which goes beyond the reward-centric or goal-centric control in other models .
* Explicit conflict arbitration: NES provides an algorithm (voting + veto) for norm conflict resolution , whereas other architectures leave value conflicts implicit or resolve them ad hoc (if at all).
* Recursive self-reflection: NES builds in a limited recursive review (RAA) that allows “second thoughts,” implementing a form of rational reconsideration loop. Traditional architectures like ACT-R or even HRL typically execute a policy without such a discrete self-reflective pass (though they can be made to re-plan, it’s not as intrinsic as NES’s design).
* Integration of emotion: Rather than treat emotion as external or ignore it, NES weaves it into decision parameters systematically . This hybrid of cold control with hot biases is often discussed conceptually (e.g. in dual-process or somatic marker theories) but NES gives it form in computations.
* Homunculus avoided through mechanism: Many earlier executive models were criticized for positing a “little man” (SAS’s homunculus, or a central executive with undefined operations). NES counters this by breaking the executive role into mechanistic components (accumulators, threshold modulators, conflict voters) . The homunculus problem is addressed head-on: NES’s “ruling faculty” is emergent from these parts, not an unexplained decider .
* Implementability: NES is specified at a level where it can be directly turned into a simulation or AI agent . As demonstrated with the Stroop task implementation, it’s not just a metaphor but an algorithm. Many of the other frameworks (especially high-level psychological ones like dual-process or SAS) needed this kind of concrete instantiation, which NES provides.

By articulating these distinctions, we ensure NES v2 is positioned clearly in relation to established models, showing how it synthesizes ideas (like drift diffusion, conflict monitoring, hierarchical control) but also where it innovates (norm-centric control, explicit “free won’t” gating ). This not only guides readers in understanding NES’s place in the theoretical landscape but also identifies how NES can be experimentally distinguished from, say, a pure reinforcement learning agent or a standard cognitive architecture (see next section for predictions).

## **Simulation-Driven Parameter Insights**

(New section to report key findings from Stroop task simulations and their implications for NES parameters.)

We conducted simulations of the NES model on a Stroop-like task – a classic test of cognitive control where an impulse (reading a word) conflicts with a task goal (naming the ink color). These simulations provide quantitative support for NES’s mechanisms and illustrate how tuning parameters affects behavior:

* Stroop Interference Effect: In the model, when an impulse conflicted with a task norm (e.g. norm = “follow color instruction” vs impulse = “read word”), the Comparator’s evidence accumulation was slower to reach a decision. Concretely, incongruent trials (word and color differ) yielded longer decision times than congruent trials (word and color same) . For example, with baseline parameters ($w\_n=1.0$ strong norm, moderate threshold), mean reaction time was 0.787 s incongruent vs 0.651 s congruent – a clear ~20% slowing . This matches human Stroop results, validating that NES’s Comparator+Assent dynamic naturally reproduces conflict slowing. The norm ($N\_i$) opposing the word-reading impulse drags the drift rate down, requiring more time to hit threshold.
* Threshold (5HT) and Speed-Accuracy Trade-off: We varied the Assent Gate’s threshold to simulate different 5HT (serotonin) levels and observed classic speed/accuracy trade-offs . Lowering the threshold (analogous to low 5HT or an impulsive setting) made decisions much faster – in the simulation, a threshold drop from 1.0 to 0.5 cut mean RT nearly in half . However, this came at the cost of accuracy if other stabilizing factors (like norm weight) were reduced. With an extremely strong norm ($w\_n=1.0$), even low thresholds didn’t produce errors (the norm influence was dominant, preventing wrong impulses) . But when we weakened the norm weight to 0.5, the low threshold condition started producing mistakes (choosing the wrong response). Specifically, at $w\_n=0.5$ and $\Theta\_{\text{base}}=0.5$ with moderate noise, accuracy dropped as low as ~41% on incongruent trials – the impulse often “won” over the norm due to the lenient gate and noise. This demonstrates that NES can manifest impulsivity errors when normative control is not dominant, just like humans make more errors when rushing.
* Noise and Decision Variability: By increasing the noise standard deviation in the drift process (simulating internal or environmental variability), we found more stochasticity in outcomes . High noise sometimes pushed evidence over the threshold for a wrong impulse before the norm could accumulate enough counter-evidence, especially under low threshold conditions. This aligns with the idea that even a person with strong values can make occasional lapses if decision threshold is low and there’s a lot of “neural noise” or distraction. NES’s stochastic accumulator captures that: errors were rare at $\sigma=0.1$ (10% noise) but became frequent (nearly 60% errors) at $\sigma=0.3$ with a low threshold . This underscores that parameter $\sigma$ (and by extension attention stability) critically affects how robust the system is against temptations or conflicting signals.
* Norm Weight as a Control Dial: The simulations confirm that $w\_n$ (normative weight) serves as a moral control dial. With high $w\_n$, the model showed strong cognitive control: even under pressure (low thresholds, noise) it mostly adhered to the norm (100% accuracy until we severely lowered both threshold and $w\_n$) . With lower $w\_n$, the model became more behaviorally flexible but prone to errors when the gate was lax . In other words, $w\_n$ modulates where the model lies on the spectrum from principle-driven restraint to stimulus-driven responsiveness. This is a testable prediction: individuals or agents with stronger internalized norms should show less influence of speed pressure on error rate (they won’t cave to impulsivity easily), whereas those with weaker norms show classic speed-accuracy trade-off curves .
* Parameter Interactions: The sweep data (thresholds 1.0 vs 0.8 vs 0.5 crossed with noise 0.1, 0.2, 0.3) in the incongruent condition illustrate an interaction: at high norm strength, threshold changes affected RT but not accuracy (ceiling performance), whereas at lowered norm strength, threshold and noise together determined accuracy. This suggests in NES there are regimes: a norm-dominated regime where behavior is rule-following and errors minimal (only speed varies), and a performance regime where if norm influence drops, the agent exhibits the classic speed-accuracy trade-off of cognitive control. NES can transition between these by tuning $w\_n$. This might mirror human behavior differences between tasks with deeply moral stakes (where people err less even if hurried) versus trivial conflicts (where speeding up causes more mistakes).

These findings, while coming from a simplified Stroop model, lend confidence that NES’s design produces realistic cognitive effects. Importantly, they ground the choice of certain parameter ranges for Blueprint v2: We learned that a base threshold around 1.0 with noise around 0.1–0.2 is a balanced setting (giving ~5–800 ms decision times and few errors in moderate conflict) . We also saw that extremes (very low threshold + weak norms + high noise) push the system to breakdown (lots of errors), which maps to impulsive pathological cases. Thus, NES v2 can recommend default parameters (e.g. $w\_n$ initially around 1.0 per strongly internalized norm, $w\_s$ around 0.5, threshold baseline 1.0) and caution that lowering normative influence too far can yield undesirable performance, unless that’s intended for a specific simulation of, say, a less socialized agent.

In future work, we will extend these simulation tests to other domains (e.g. moral dilemmas, multitasking scenarios) to further chart how NES parameters influence outcomes. The Stroop case demonstrates the model is quantitatively predictive and aligns with known human data (conflict cost, speed-accuracy trade-off), a strong validation point for NES’s core mechanisms .

## **Experimental Predictions and Evaluation Plans**

NES v2, with its detailed specifications, yields several falsifiable predictions that can be tested in human experiments or AI simulations. We outline some concrete predictions:

1. Serotonergic Modulation of Response Caution: NES predicts that increasing serotonin levels corresponds to higher decision thresholds, leading to slower but more accurate (norm-compliant) decisions . Prediction: If human participants are given a drug that increases 5HT (like an SSRI) or if they are in a state associated with high serotonin, they should exhibit longer reaction times in impulse control tasks and fewer impulsive errors. Conversely, low serotonin states (or acute tryptophan depletion in lab studies) should produce faster responses with more impulsive choices. This is consistent with some findings in psychopharmacology, but NES provides a specific mechanism to test: e.g., using tasks that involve moral or rule-based inhibition (not just motor inhibition) to see if serotonergic manipulation selectively increases adherence to internal rules (e.g., less cheating or violating instructions under high 5HT).
2. Norm Priming and Strength Effects: Because $w\_n$ represents the strength of internalized norms, NES predicts that priming an individual’s norms will alter their behavior in conflict situations. Prediction: If we remind someone of a strong moral norm (e.g. have them read a moral creed or think of an honor code) before a decision task that pits that norm against temptation, their effective $w\_n$ will be elevated and they will make more norm-consistent choices (even under time pressure) than a control group . For instance, ask participants to either recall an honesty-related oath or not, then present them with an opportunity to lie for monetary gain under time pressure. NES would predict the primed group lies less and perhaps takes a bit longer to respond (as their comparator gives more weight to honesty norm), whereas the unprimed group shows a higher rate of quick, self-serving lies. This is testable and goes beyond generic dual-process theory by quantifying the effect of norm salience.
3. Emotional Hot/Cold Influence on Decision Thresholds: NES posits that emotional arousal has immediate effects on thresholds . Prediction: Inducing a hot emotional state (like anger or excitement) in subjects will lower their effective decision thresholds in a subsequent unrelated task, observable as faster reactions and potentially riskier choices. For example, after being insulted (to induce anger), participants might complete a Go/No-Go or Stroop task – NES predicts faster responses and more errors (especially errors that align with prepotent impulses) compared to a neutral emotional state. A cold state induction (e.g. sadness or calm focus via meditation) should do the opposite – slower, more cautious responses (higher threshold). This could be measured via cognitive tasks or even moral decision-making scenarios (anger might make people more punitive and quick to judge, simulating NES’s lowering of thresholds and norm biases like “justice” over “mercy” ).
4. Conflict-driven Recursive Deliberation (RAA evidence): If NES’s RAA mechanism is accurate, we should see signs of “thinking twice” in human data when conflicts are tough. Prediction: In difficult moral dilemmas or high-conflict trials, people might exhibit a brief pause or a double-peaked reaction time distribution, as if they started to respond, reconsidered, and then finalized a decision. This could be tested by tracking not just RT but perhaps neural markers: NES suggests that in a hard conflict, ACC (conflict monitor) activity would spike, and additional frontal activity would occur as the person engages in a secondary evaluation (RAA invoking control). Empirically, one might use EEG or fMRI to see if a second-stage control signal appears in trials where initial impulses were misleading. Behaviorally, one could examine sequences: if a subject initially leans one way (e.g., changes their mind mid-response), that aligns with NES’s prediction of a recursive check causing a reversal. Standard one-shot models (like diffusion without collapse or a one-layer policy) wouldn’t predict such mid-course corrections as naturally.
5. Norm Conflict Resolution Outcomes: NES’s Norm Conflict Resolver yields specific outcomes – e.g., if two norms conflict, whichever has higher weight or veto status wins. Prediction: If we set up a scenario where participants have to choose between two valued principles (say fairness vs loyalty), we can predict outcomes by independently measuring how strongly individuals hold each principle (through surveys or prior behavior). NES would predict that the principle with the higher subjective weight will guide the decision, unless the other principle is considered inviolable by that individual (veto-like). This could be tested by correlating people’s known value rankings with their choices in novel dilemmas. If someone ranks loyalty far above fairness generally, NES would predict they choose the loyal action even when it seems unfair, and do so with less conflict (shorter RT) than someone who values both equally (who will experience a tie and longer deliberation). This is a way to validate the weighted voting model in human ethics: do personal value weights predict resolution of moral conflicts? NES suggests yes – essentially a quantitative model of moral decision-making.
6. “Free Won’t” – Deliberate Withholding: NES incorporates a conscious veto mechanism akin to Libet’s “free won’t” . Prediction: In tasks where a prepotent action is triggered but a late stop is possible (like the classic Libet task or stop-signal tasks with moral stakes), we should see evidence of a distinct veto process. Specifically, some trials where an action was nearly executed will show a sudden inhibition at the last moment, which NES attributes to the Assent Gate denying final approval. Neural prediction: a surge in control-related brain activity (right inferior frontal cortex or basal ganglia STN involvement, known in stopping) right before the action would have been taken, even though the action is not taken. Behaviorally, this might show up as partial responses aborted (like someone starting to press a button then stopping). NES provides a framework to quantify how often and under what conditions this veto occurs (e.g., when a strong norm kicks in slightly slower than an impulse – perhaps seen if we tell subjects “press this button quickly for reward, unless you see a moral rule violation warning”). The presence of such last-instant vetoes, especially in moral contexts, would support NES’s decision gate concept over models that lack an explicit final check.
7. Adapting to New Norms: Because NES can learn and adjust norms, we predict certain learning trajectories. Prediction: If an agent or person is put in a new environment with a new rule, initially their behavior might violate that rule until the Norm Repository updates through feedback . Over time, mistakes should decrease as the norm weight increases. For example, if we introduce participants to a game with an unusual norm (“never take the last coin from the pot”) and enforce it with mild penalties, NES would predict a learning curve: first few rounds, many take the last coin (impulse for reward wins), after several penalties, they start conforming more (norm weight rises). The model could fit the curve of errors over trials by adjusting a simulated $w\_n$. If the data fits the model’s update rule (e.g., a Bayesian or reinforcement learning update of norm strength), that’s evidence for NES’s learning component. Likewise, if the penalty is removed (norm no longer enforced), NES predicts a slow decay in compliance over time , which can be observed experimentally as people revert to more selfish behavior gradually when they notice no consequences.

Each of these predictions maps to a measurable outcome, allowing NES v2 to be empirically evaluated. The Stroop simulation already confirmed some basic predictions (e.g., conflict slows responses; threshold modulation trades speed for accuracy) . The above go further into the realm of complex behavior, emotion, and learning. Our plan is to design human experiments and AI agent simulations for these scenarios. Success would be measured by NES’s ability to predict not just qualitative trends but quantitative details (e.g., RT differences, error rates, correlation strengths between value weights and choices). By validating these, we would establish NES as a robust model of executive function. Conversely, if any prediction fails (say, emotion induction doesn’t affect threshold as expected), that will guide revisions to the model (perhaps adding nuance to how emotions are represented).

In sum, NES v2 is not just a theoretical construct – it’s a framework yielding clear hypotheses about decision-making. This makes it falsifiable and therefore scientifically valuable: experiments can in principle refute or support the existence of a normative gating process, or the particular way recursion is used, etc. Through iterative testing and model refinement, NES can be honed to accurately mirror the human normative executive, or identify which components need adjustment (e.g., maybe real norms don’t “vote” linearly – a failed prediction there would be informative). This commitment to empirical grounding strengthens NES against criticisms of being merely philosophical.

## **Addressing Potential Critiques and Risks**

Finally, we bolster Blueprint v2 by explicitly countering major conceptual concerns:

* Homunculus Critique: One might argue NES’s RAA (meta-controller) is just a “little man in the head” making decisions, deferring the problem. We counter that by pointing to the fully mechanistic decomposition of the executive . NES doesn’t have an infinite regress of deciders; it has one top-level loop (RAA) that is finite and rule-bound, and all operations within it are mechanistic (drift diffusion, threshold comparison, weighted voting). The RAA is termed “Agent” metaphorically, but it’s essentially a set of if-then conditions and adjustments – much like a thermostat (simple control logic) rather than a sentient mini-agent. Blueprint v1 used the analogy of banishing the homunculus by an “army of idiots” – simple processes that together do the work . We retain that spirit: each component (comparator, gate, resolver) on its own is dumb and has no homuncular understanding; it just follows simple rules (accumulate evidence, etc.). It is their interaction that produces what looks like rational agency. Furthermore, the recursion stops based on fixed criteria (cycle count or urgency) , so there is no need for a ghostly “decider to stop deciding” – it’s baked into the logic. Thus, NES avoids the infinite regress by design and is fully compatible with mechanistic neuroscience perspectives where executive function emerges from coordinated networks, not a single mysterious locus .
* Cultural and Bias Concerns: NES contains an explicit Norm Repository, which means the system’s behavior is only as good as the norms it has internalized. A worry is that this could hard-code cultural biases or unethical norms if those are learned. We address this on two levels:  
  + Norm Transparency and Adjustability: Because norms are explicit in NES, one can inspect and modify them. This is actually an advantage for AI alignment and bias mitigation – unlike a black-box neural net that might carry biases invisibly, NES would have its norms laid out (with weights, tags like source, context, etc.). If a bias is present (say the agent learned a prejudiced norm from bad data), it can be identified and corrected either by further training (the agent experiencing counterevidence or a re-training of that norm’s weight) or by designers intervening (since this is a high-level declarative element). In human terms, NES’s account suggests even deeply internalized biases (norms) can be confronted by reflection or new learning, as the architecture permits norm revision. This aligns with how people can overcome cultural biases through education – NES can simulate that process (not simply act on biases unchangeably).
  + Canonical Alignment Guide: We assume the implementation of NES, especially in AI, would be accompanied by a Canonical Alignment Guide (as noted in the project context) that defines which norms should be instilled and how to ensure they are ethically sound. NES’s structure is neutral to which norms it carries – it could run with malevolent norms too. So the responsibility is to seed it with a set of norms aligned to ethical principles and to have mechanisms for norm evaluation (meta-norms) that can flag inconsistent or harmful norms. For instance, a meta-norm could be “if a norm causes unjustified harm, eventually reject that norm,” giving the system an internal corrective for norms that produce bad outcomes. Culturally, this means NES can adapt to improved norms over time (e.g., if society moves away from a prejudice, the agent’s norms decay for that behavior). We emphasize that NES is a framework to hold norms – it does not itself judge the content. But its design encourages oversight: one could audit the Norm Repository, unlike in end-to-end learning systems. Thus NES can be seen as a tool for implementing ethical governance in AI, provided the initial norm set and learning environment are carefully aligned with fairness and human values . In sum, rather than amplify bias, NES actually externalizes and potentially attenuates it through transparency and the possibility of recalibration.
* Free Will and Determinism: Some might claim that by turning decision-making into an algorithm, NES implies there is no free will, or conversely that it is smuggling in free will via the RAA. Our stance is that NES is agnostic about philosophical free will but provides a model for the psychological experience of will. It implements a “Free Won’t” veto as Libet envisioned – meaning the system can overrule an impulse at the last moment, which is a cornerstone of what many consider conscious will. If one defines free will in a compatibilist sense (the ability to act according to one’s reasons and values), NES actually exemplifies it: the agent’s actions are indeed governed by its internal reasons (norms, goals) rather than just reflex. The system can do otherwise in a given situation if its values say so (it’s not locked into a single policy). Moreover, NES allows for reflective self-modification (through norm updates and RAA recursion), which resonates with how humans can reflect on their motives and change. This is arguably a basis for autonomy. Certainly, NES is deterministic (or stochastic under noise) in an implementation sense, but so are biological brains to a large extent. The important point is NES can still differentiate between an unfree agent (one with no normative control, just acting on impulse or external control) and a “free” behaving agent (one that exercises internal veto and guidance). If an agent had no Normative Executive, it would simply chase reward or habit; with NES, it can say “I won’t do X because I believe it’s wrong,” which is a hallmark of exercising will. To directly address skepticism: NES doesn’t prove or disprove metaphysical free will, but it does offer a model for the functionality often attributed to free will (rational self-control). It shows how a system can be completely naturalistic and yet exhibit what we recognize as volitional acts. This can soften free-will skepticism by bridging the gap – showing that what we call willpower or moral choice need not be a mystery, but can be the output of a well-specified process . In practical terms, if someone argues “the executive just follows norms, so where’s freedom?”, we answer: the origin of those norms and the ability to question them is where freedom lies. NES allows norms to be questioned (through RAA meta-evaluation and norm learning) – thus the agent isn’t slavishly following an unalterable code, it is actively curating its code over its life. This aligns with a view of free will as the capacity for self-governance and moral responsibility.
* Complexity and Feasibility: Another concern could be that NES is overly complex or unrealistic neurobiologically. In response, we note that each piece of NES maps onto known brain processes (though NES is an abstract functional model, not a neural net):  
  + The Comparator resembles integrator circuits in the brain that accumulate evidence for decisions (linked to parietal and frontal areas in decision tasks).
  + The Assent Gate might map to mechanisms in the subthalamic nucleus and prefrontal cortex that adjust decision thresholds under modulation (the STN is implicated in holding a decision when conflict is detected – effectively raising a threshold – which fits our model’s logic).
  + The Norm Repository corresponds to long-term memory networks encoding values and rules (likely distributed in prefrontal and temporal cortices).
  + The Conflict Resolver reflects functions of frontal polar cortex or deliberative reasoning processes that weigh abstract rules (and perhaps the ACC in mediating between competing rule-based responses).
  + The RAA aligns with the notion of meta-cognitive control, possibly implemented by frontal networks monitoring performance (like the medial prefrontal cortex/ACC and lateral prefrontal interplay).
  + Emotion integration comes via neuromodulators (serotonin, dopamine for reward vs norm conflict perhaps, norepinephrine for urgency) and limbic signals (amygdala for threat causing threshold changes, etc.), all of which there is precedent for in cognitive neuroscience.
* NES doesn’t violate what’s known – it packages it in a specific architecture. Yes, it’s more detailed than a simple “one module” model, but the brain itself is complex. The design is actually an advantage computationally: it is modular and thus more tractable to implement or modify than a monolithic black box. The success of our Stroop simulation (a relatively small model capturing key effects) shows that NES’s complexity is manageable. We aim to progressively test and refine each part, and if any part proves too unwieldy, that will emerge in testing (thus far, it hasn’t).

In conclusion, Blueprint v2 addresses these possible objections with a combination of theoretical reasoning and empirical alignment. NES is positioned as a transparent, adjustable, and empirically grounded model of the executive, rather than a mystical or inflexible one. By preemptively discussing the homunculus issue, cultural norm biases, and the free will question, we make it clear that NES is aware of its philosophical implications and is constructed to be on solid scientific and conceptual ground. Each module’s inclusion is justified by either data or clear necessity, and nothing supernatural is assumed. We believe this rigorous approach will assuage critics who might have seen NES v1 as too conceptual – in v2, everything is spelled out to show that the Normative Executive System is both implementable and aligned with what is known about minds and machines.

\*\*\* (All revisions above adhere to the tone and principles of the Canonical Alignment Guide, ensuring consistency in terminology and a balanced integration of philosophical insight with technical detail .)\*\*\*