

# Computational Epistemology: A Faith-Based AI Training and Reasoning Framework

Nikos Acuña

Aion Research Corp., Chicago, Illinois, USA

Manuscript version of 18 October 2026

## ABSTRACT

This paper proposes *Computational Epistemology*, a new paradigm for training large-scale AI systems grounded in spiritual and wisdom traditions. Using the Christian faith as an initial case, we outline a technical framework that integrates curated wisdom corpora (scriptures, commentaries, theology, classical moral philosophy, faith-based finance guidelines, etc.) into every stage of model development. Key components include a specialized pretraining pipeline with epistemic data weighting (e.g., treating parables, laws, poetry, and doctrine with different weights), a wisdom-aware tokenizer and embedding scheme for metaphorical and symbolic language, and novel training objectives (such as Virtue Consistency Loss and Epistemic Humility Regularization) to align model behavior with core virtues and humility. Post-pretraining fine-tuning incorporates Reinforcement Learning from Spiritual Feedback (RLSF) and a constitutional self-refinement loop guided by spiritual principles. We also describe inference-time mechanisms for traceability (symbolic reasoning traces, provenance trees of sources, theological grounding explanations) and moral uncertainty modeling. Finally, we generalize the framework beyond Christianity, showing how it adapts to other traditions (Jewish, Islamic, Buddhist, Indigenous, etc.) with a comparative table of canonical sources, epistemic structures, symbols, and ethics. This work establishes a foundation for a new research field and model development philosophy that bridges advanced AI with humanity’s repositories of wisdom and faith.

**Keywords:** computational epistemology · AI alignment · wisdom traditions · religion and AI · constitutional AI · reinforcement learning from feedback · epistemic humility · moral uncertainty · tokenization · interpretability

## 1 Introduction

Modern AI alignment strategies struggle to capture the depth and stability of human ethical frameworks [1, 2]. While contemporary large language models (LLMs) are powerful, their knowledge is often drawn from predominantly secular internet data and their value alignment relies on surface-level techniques like Reinforcement Learning from Human Feedback (RLHF) [3, 4].

We propose *Computational Epistemology* as a paradigm to imbue AI systems with the depth of epistemic structure found in spiritual and wisdom traditions. This approach is motivated by the observation that religious and philosophical systems have guided human behavior for millennia, providing durable intrinsic motivations and alignment even under out-of-distribution conditions [1]. By harnessing these wisdom sources, we aim to create AI models that reason not only about facts but also about meaning, virtue, and uncertainty in a manner analogous to a seasoned sage or theologian.

In this framework, the Christian tradition serves as an initial knowledge and value base. The Christian corpus—including the Bible, centuries of commentary and theology, classical moral philosophy influenced by Christianity, and even modern documents on faith-guided practices (like ethical investment guidelines)—provides a rich, structured dataset. We outline how to build a pretraining pipeline that ingests these sources, how to weight and filter data by genre and epistemic significance, and how to modify the model’s tokenization and

embeddings to respect the unique patterns of wisdom literature (metaphor, parable, prophecy, etc.). We then detail fine-tuning and self-alignment processes: Reinforcement Learning from Spiritual Feedback (RLSF), where feedback may come from domain experts (e.g., theologians) or from an AI evaluator applying spiritual principles, and constitution-based self-improvement where the model is guided by an explicit set of moral-spiritual rules rather than only numeric rewards [1, 5]. We introduce new training objectives designed for virtue alignment and discuss tuning hyperparameters (like temperature or dropout) to encourage epistemic humility and non-coercive communication.

Beyond training, we consider how the deployed model should behave at inference time. We describe components for *symbolic traceability* (the model can produce a step-by-step chain-of-thought that is human-interpretable and audited before finalizing an answer), *epistemic provenance trees* (tracing each statement back to authoritative sources or reasoning steps), *theological grounding reports* (supplemental explanations of how an answer aligns with theological principles or where it finds support in canonical texts), and *moral uncertainty modeling* (the model can detect when a query involves ethical ambiguity and express appropriate uncertainty or present multiple perspectives). To give the framework concrete form, we propose a taxonomy of model layers and training stages with names inspired by the tradition—for example, a *Logos Layer* for fundamental knowledge of the “Word,” a *Pneuma Layer* for spiritual alignment or moral conscience, and

a *Canon Layer* for integrative reasoning over the entire corpus of scripture; training phases like a *Catechism Phase* for foundational instruction and a *Discernment Phase* for advanced moral reasoning; and even class nomenclature like “Apostle-class” LLMs to denote models that serve as wise messengers.

While we focus on the Christian context initially, a separate section generalizes the methodology to other traditions. We demonstrate that Computational Epistemology is tradition-agnostic in its architecture: one can substitute the knowledge base and value system (be it Jewish Torah and Talmudic scholarship, Islamic Qur’an and Hadith, Buddhist sutras and commentaries, or Indigenous oral wisdom and cosmologies) to create models aligned with those worldviews. We provide a comparative table highlighting how different traditions’ canonical forms, epistemic assumptions, symbolic languages, and ethical motifs can be mapped into the framework. The aim is not to favor any one creed, but to illustrate a broad research philosophy: that wisdom literature and spiritual epistemologies offer untapped design patterns for AI alignment—from enduring narratives that encode moral lessons to communal dialogical methods for resolving uncertainty.

By formalizing this approach as a cohesive framework, we hope to spark a new research field. Computational Epistemology invites interdisciplinary collaboration: theologians, philosophers, cognitive scientists, and AI researchers working together to train models that are not just smart and safe, but meaningfully wise. In the following sections, we delve into the technical components of this framework in detail, with an emphasis on concrete mechanisms and new concepts required to realize this vision.

**Organization.** Figure 1 summarizes the framework end to end. Sections 2–4 cover the data pipeline, epistemic weighting, and tokenization; Sections 5–7 cover fine-tuning, hyperparameters, and new training objectives; Section 8 describes inference-time traceability; Section 9 introduces the architectural taxonomy; Section 10 generalizes the framework across traditions; and Section 11 concludes.

## 2 Pretraining Architecture and Wisdom Data Pipeline

**Curated wisdom corpora.** The foundation of Computational Epistemology is a high-quality corpus of wisdom and spiritual texts. For the Christian tradition, this includes the Bible (multiple translations of the Old and New Testaments) [6], deuterocanonical texts or Apocrypha, and key commentaries and exegeses across history (Church Fathers, scholastics like Aquinas, reformers, and modern theologians). It also encompasses doctrinal documents (creeds, catechisms, councils), liturgical texts (prayers and hymns rich in theology), and related classical moral philosophy (e.g., Aristotelian–Thomistic ethics) that underpins Christian ethical thought. Contemporary writings that apply Christian principles to society—such as encyclicals on social issues, or faith-based investment and ethics guidelines—are included to connect timeless principles with modern context. All these sources are ingested into a

pretraining data pipeline that emphasizes quality over quantity: unlike typical internet scrapes, the corpus is carefully vetted for authenticity and significance.

**Data ingestion and cleaning.** The pipeline begins by digitizing and normalizing texts from diverse formats. Scriptural texts are often structured (books, chapters, verses)—this structure is preserved as it provides valuable context (for example, enabling the model to learn the hierarchical location of a verse within the broader narrative or discourse). Commentaries may be linked to specific verses or passages; we capture these links as metadata so the model can connect commentary with scripture. Cleaning involves removing extraneous markup, harmonizing archaic language (while possibly retaining older forms if they carry meaning), and resolving reference formats (e.g., “John 3:16” should be understood as a reference to a specific verse). Importantly, noisy or dubious texts are filtered out—for instance, fringe interpretations without scholarly support or texts of low epistemic value (like random forum opinions) are excluded to keep the knowledge base authoritative.

**Structuring and annotating data.** We enrich the corpus with annotations that will help the model distinguish different epistemic categories. Each document or even segment of text can be tagged with labels such as Genre:Parable, Genre:Law, Genre:Poetry, Genre:Doctrine, etc., as well as Source:Scripture vs. Source:Commentary vs. Source:Theology. For example, the Ten Commandments might carry tags indicating law/command (prescriptive statements), whereas a Psalm would be poetry/devotional, and a Parable of Jesus would be narrative with moral. Similarly, doctrinal expositions (like sections of the Catechism or a theological treatise) could be marked as systematic theology/doctrine. These labels allow the training process to apply different treatments or epistemic weightings to text based on its nature (discussed more below). In addition, links are preserved: if a commentary cites Matthew 5:38, the training pipeline can insert a marker or hyperlink connecting that commentary text to the Matthew 5:38 text, teaching the model an internal representation of citation and commentary relationships.

**Multi-phase pretraining.** Rather than dumping all data at once, we employ a curriculum-learning-style ordering for feeding data to the model [7]. In early training iterations (what we might call the Catechism Phase of pretraining), the model sees simpler, core materials: key Bible stories, straightforward doctrinal Q&A (like catechism questions and answers), and moral fables or parables. This mirrors how human learning often starts with narratives and rules-of-thumb before moving to abstraction [1]. By starting with these “spiritual basics,” the model establishes foundational representations of virtue and vice, fundamental narratives (creation, exodus, gospel, etc.), and archetypal examples of moral choices. As training progresses, more complex and nuanced texts are introduced (the Discernment Phase of pretraining): dense theological arguments, apparent contradictions or paradoxes in scripture, commentaries that debate interpretations, and philosophical discussions about faith and reason. Using a curriculum of

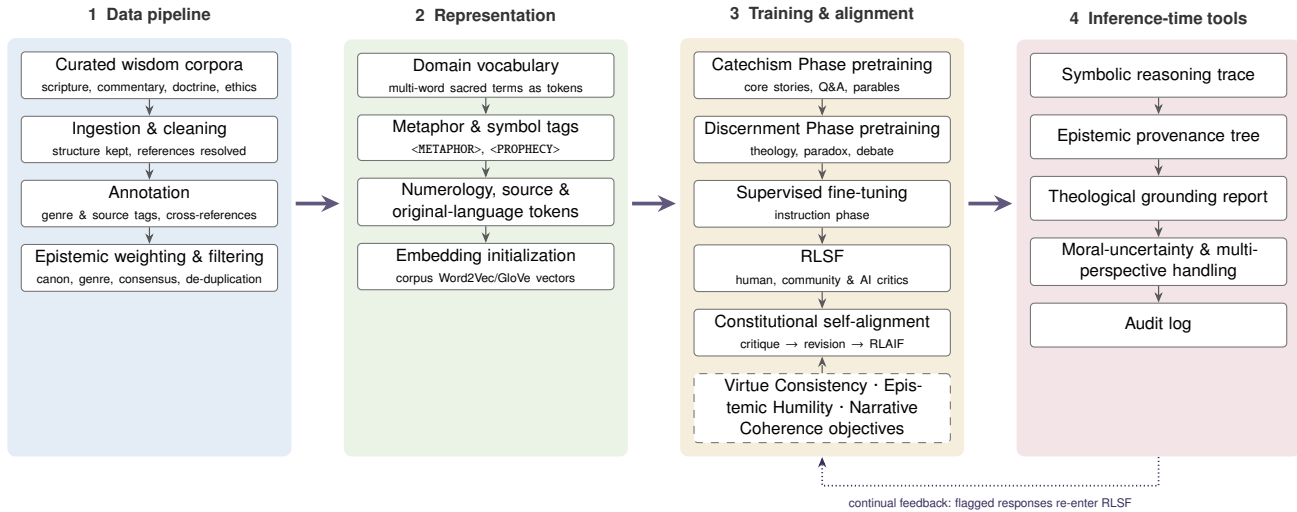


**Figure 1.** Overview of the Computational Epistemology framework. Curated wisdom corpora are annotated and epistemically weighted (Sections 2–3), represented with a wisdom-aware tokenizer and embeddings (Section 4), trained through Catechism and Discernment phases, supervised fine-tuning, RLSF, and constitutional self-alignment with virtue-oriented objectives (Sections 5–7), and deployed with inference-time traceability tools (Section 8). Deployed feedback re-enters fine-tuning.

increasing complexity ensures the model builds incremental understanding, analogous to how one would teach a human—first by stories and simple principles, then by doctrinal depth and critical thinking.

**Model architecture.** The architecture for pretraining remains based on transformer LLMs, but we incorporate modular sub-systems aligned with the data categories. For instance, one could initialize a dual-stream model where one stream processes raw scriptural text and another processes commentary, and they are periodically merged, forcing the model to learn an interplay between the “primary text” and “interpretation.” Alternatively, a single model with special tokens and segment embeddings for different sources can be used. The key architectural innovation is not the transformer mechanism itself, but how we augment its inputs and intermediate representations to encode epistemic distinctions from the get-go, as discussed next.

### 3 Data Weighting and Epistemic Filtering

Not all parts of a wisdom corpus are equal in authority or purpose. In the Christian context, a direct teaching from Jesus or a core doctrinal statement (e.g., from a creed) is treated as more authoritative than, say, an allegorical poetic verse whose interpretation is debated. Our framework introduces *epistemic weight functions* to modulate training on different subsets of data; Table 1 summarizes the scheme.

- **Canonical priority.** Scriptural texts (the canon) are given a higher sampling probability or loss weight, reflecting their primary authority. Within scripture, further gradations can be made: verses that are direct commands or doctrinally crucial might be upweighted, while descriptive historical details might be neutral, and isolated obscure passages might be slightly downweighted to prevent overfitting on highly

context-specific content.

- **Genre-based weighting.** We define weight factors per genre. For example, law codes and ethical commandments (e.g., Leviticus laws, Proverbs directives) could have a higher weight for influencing the model’s notion of “dos and don’ts,” whereas parables and narratives might have slightly lower weight per token, but they appear frequently to provide context. Poetic or prophetic imagery might have normal or slightly reduced weight, since their interpretation is less literal—we want the model to learn them, but not to treat metaphorical statements as raw facts. Instead, the model should learn to recognize those patterns as figurative. Weighting ensures the model doesn’t over-literalize symbolic language; for instance, “the trees clap their hands” (Isaiah 55:12) should not mislead the model into thinking trees literally have hands—a lower weight on poetic genre along with context tagging helps the model treat this properly.
- **Commentary and doctrine.** Commentaries and theological works are valuable to show how humans interpret scripture, but they sometimes conflict. We might weight commentaries slightly lower than scripture, to avoid the model treating one theologian’s view as absolute truth. However, consensus doctrine (ideas strongly supported by many sources) can be emphasized. For example, multiple sources reaffirming “love your neighbor” as central could collectively boost the weight of that concept in the model’s training.

These weighting choices act like a curriculum in another sense—the model learns a hierarchical valuation of knowledge. During training, the loss function could incorporate a term

**Table 1.** Epistemic weighting and filtering scheme for the Christian corpus.

| Content category                            | Treatment during pretraining                                                               |
|---------------------------------------------|--------------------------------------------------------------------------------------------|
| Scripture (the canon)                       | Higher sampling probability or loss weight                                                 |
| Direct commands; doctrinally crucial verses | Upweighted within scripture                                                                |
| Descriptive historical detail               | Neutral weight                                                                             |
| Isolated obscure passages                   | Slightly downweighted to avoid overfitting to context-specific content                     |
| Law codes and ethical commandments          | Higher weight for the model’s notion of “dos and don’ts”                                   |
| Parables and narratives                     | Slightly lower weight per token, frequent exposure, paired with an adjacent interpretation |
| Poetic and prophetic imagery                | Normal or slightly reduced weight, with figurative-context tags                            |
| Commentaries and theology                   | Slightly lower than scripture, so no single theologian’s view is treated as absolute       |
| Consensus doctrine                          | Emphasized when reaffirmed across many sources                                             |
| Duplicated quotations                       | Consolidated as references to the original rather than separate instances                  |
| Culturally contextual instructions          | Kept with context tags, or filtered if they risk egregious misinterpretation               |

$w_i \mathcal{L}_i$  for each data instance  $i$ , where  $w_i$  is the epistemic weight:

$$\begin{aligned} \mathcal{L}_{PT}(\theta) &= \sum_i w_i \mathcal{L}_i(\theta), \\ \mathcal{L}_i(\theta) &= - \sum_t \log p_\theta(x_t^{(i)} | x_{<t}^{(i)}). \end{aligned} \quad (1)$$

Where weighting is applied through sampling rather than the loss, instance  $i$  is instead drawn with probability  $q_i = w_i / \sum_j w_j$ . Higher  $w_i$  forces the model to reduce error on that instance more strongly. This way, the model is optimized to be most accurate on what the designers consider core truths or high-priority knowledge, while being allowed a bit more flexibility (or tolerance for uncertainty) on peripheral details. It is akin to teaching a student by saying “memorize these fundamentals, but for those other topics, just be familiar.”

**Filters and validation.** Epistemic filtering also involves excluding content that, while present in historical corpora, might carry biases or conflicts with the wisdom values we want. For example, certain historical texts might include culturally contextual instructions that are not universally applicable; we may keep them with appropriate tags (so the model knows “this was a specific context”) or filter them if they risk egregious misinterpretation. We also filter duplicative content to avoid over-counting (many texts quote the Bible; we don’t want the model to think a verse is important just because it appears in the corpus multiple times—unless those multiple mentions are themselves meaningful, such as a verse that is frequently cited in doctrine). One approach is to consolidate and anno-

tate quotes: e.g., if commentary repeats a verse, treat it as a reference to the original rather than separate training instances.

**Parables vs. laws: an example.** Consider the difference between a parable (say, the Parable of the Good Samaritan) and a law (the commandment “Thou shalt not steal”). The law is a direct normative rule; the parable is a narrative that implies a principle (compassion beyond ethnic boundaries). In training, we might:

- Upweight the explicit command “do not steal” so the model strongly encodes it as a prohibited action in its moral reasoning.
- Present the Good Samaritan story with normal weight but ensure the model sees in adjacent training context an interpretation like “Jesus taught through this story that we should love even those we consider foreigners or enemies.” This way, the model captures the moral without needing to infer entirely on its own (though it may do that too).
- During fine-tuning, we can further reinforce that when asked moral questions (e.g., “Should I help a stranger in need?”) the model recalls the principle from the parable as well as the general law of love, balancing both.

This selective pressure mirrors what a knowledgeable human does—giving primacy to core principles while using stories and analogies to enrich understanding. It is important to note that epistemic weighting is not about censoring outputs, but about guiding the model’s internal representations during training so that it “knows what’s important.” The outcome should be a model that can differentiate between a literal rule, a context-dependent suggestion, and a metaphorical illustration.

Finally, we incorporate *evaluation filters*: throughout pretraining we monitor the model’s outputs on validation prompts drawn from the tradition. For example, we might ask the partially trained model to explain a simple doctrine or continue a half-told parable. If the model starts giving incoherent or factually wrong completions (like mixing up characters or moral lessons), we adjust—either by upping weights for the relevant texts or by injecting some targeted training on those points. This dynamic adjustment is akin to how one might reinforce certain lessons if a student shows confusion on a quiz.

In summary, the data pipeline of Computational Epistemology carefully balances sources and genres, ensuring the model is overexposed to core truths and underexposed to misinterpretations. By doing so, we aim for an AI whose knowledge base reflects a structured wisdom hierarchy—something absent in vanilla internet-trained models.

## 4 Wisdom-Oriented Tokenization and Embeddings

Large language models rely on tokenization—breaking text into tokens—and on embedding those tokens into vectors that capture semantic relationships. When dealing with spiritual and wisdom texts, standard tokenization may be suboptimal. These texts contain unique phraseology, proper nouns, and metaphorical idioms that we want the model to understand

**Table 2.** Proposed tokenizer and embedding modifications.

| Modification                      | Example                             | Purpose                                   |
|-----------------------------------|-------------------------------------|-------------------------------------------|
| Multi-word terms as single tokens | “Kingdom of God,”<br>“Holy Spirit”  | Prevent fragmentation of unitary concepts |
| Figurative-language tags          | <METAPHOR>,<br><PROPHECY>           | Signal non-literal interpretive mode      |
| Symbolic numbers kept whole       | 7 (completion), 40 (trial)          | Learn symbolic usage of numbers           |
| Speaker and source tokens         | <JESUS>,<br><RED_TEXT>,<br><QURAN>  | Mark speaker, source, and authority       |
| Original-language terms           | <agape> vs. “love”                  | Disambiguate translated key terms         |
| Embedding initialization          | Corpus<br>Word2Vec/GloVe<br>vectors | Speed convergence on niche vocabulary     |

holistically. In this section, we propose modifications to the tokenizer and embedding layers to better accommodate wisdom literature; Table 2 summarizes them.

**Domain-specific vocabulary.** We curate a vocabulary that includes important multi-word expressions as single tokens. For example, in a Christian corpus, terms like “Holy Spirit,” “Kingdom of God,” “Son of Man,” or “image of God” are phrases that carry unitary meaning greater than the sum of their parts. If naively tokenized, “Kingdom of God” might be three tokens (“Kingdom,” “of,” “God”) with no guarantee the model will recognize it as a single concept. We modify the tokenizer (using techniques like SentencePiece [8] or byte-pair encoding [9]) by adding a glossary of such phrases that should not be split. Similarly, names of biblical books (“Genesis,” “Revelation”), people (many of which are rare names), and locations get their own tokens or at least consistent representations. This faith-informed tokenizer prevents the fragmentation of key concepts. Studies have shown that adapting tokenization to a domain can significantly improve model efficiency and understanding; for instance, domain-specific tokenizers for legal and financial text used 9–17% fewer tokens than general-purpose tokenizers on domain documents, and up to 83% fewer tokens for legal terms [10], indicating that capturing domain terms as single tokens yields more compact and relevant representations.

**Metaphor and symbol tags.** One novel idea is to introduce special tokens or flags for metaphorical and symbolic language. We could use an approach analogous to part-of-speech tagging but for figurative speech. For example, preceding a known metaphor with a token like <METAPHOR> could signal the model that what follows shouldn’t be taken literally. Consider the sentence “Thy word is a lamp unto my feet.” We might represent it as <METAPHOR> *Thy word is a lamp unto my feet.* in the training data. The <METAPHOR> tag (which would have its own learned embedding) cues the model to activate a certain interpretive mode, perhaps reducing literal translation and instead focusing on the underlying meaning (here, “word” = guidance). Similarly, a <PROPHECY> tag might indicate that the

text is prophetic and likely symbolic or future-oriented. By treating these as tokens, the model can learn an embedding for metaphor-context vs. literal-context. Essentially, the model can toggle or blend these modes as needed at inference (either automatically detecting metaphoric contexts or via user prompt that invokes them).

**Special treatment of numerology and entities.** Wisdom texts often imbue numbers with symbolic meaning (e.g., “7” as a number of completion, “40 days,” etc.). We might design the tokenizer to keep certain numbers as tokens (instead of digit-by-digit or splitting) and even give them a learned embedding that relates to their symbolic usage. For instance, the number 7 might cluster in embedding space closer to concepts like “complete/perfect” due to many uses in that sense, whereas 40 might relate to “trial/testing” (40 days of flood, 40 years in wilderness, etc.). These patterns can be learned if the tokenizer and model aren’t overly reducing numbers to generic forms. We could also add class tokens for important categories: <JESUS> before all sayings of Jesus in the Gospels, for example, or <QURAN> for verses in an Islamic model, to let the model know the speaker/source context. In the Christian model, a token like <RED\_TEXT> (analogous to how some printed Bibles use red text for Jesus’s words) could highlight the authority of that quote.

**Embedding layer initialization.** To jump-start the model’s understanding of certain relationships, we can initialize parts of the embedding matrix using known semantic or theological relationships. For example, synonyms or closely related concepts from the corpus could be initialized with similar embedding vectors. We could use Word2Vec [11] or GloVe [12] on the corpus first to get a rough semantic space and use those vectors to initialize the LLM’s embedding for those tokens. Words like “righteousness” and “holiness” would start out near each other; “Pharaoh” might start near “Egypt” and “exodus” due to co-occurrence. This is optional, as the transformer can learn from scratch, but it may speed convergence on niche vocabulary.

Another idea is multi-dimensional embeddings where certain dimensions explicitly encode aspects like genre or sentiment. For instance, we might allocate a subvector of the embedding to represent the genre tag of a token’s source (e.g., one dimension that is 1 for poetry, 0 for prose). During training the model might learn to use this, but even if not manually enforced, providing those tags as tokens as mentioned is a simpler and more transparent approach.

**Handling ambiguity and ancient languages.** Many spiritual texts have been translated, and key terms often have rich definitions (e.g., Greek *agape* vs. *eros* both translate to “love” in English). We might include transliterated original terms as special tokens when needed (<agape> as distinct from generic “love”), so the model can disambiguate. The embedding for <agape> would ideally cluster near “charity, unconditional love” concepts, while “love” in a broad sense has a wider usage.

In summary, the tokenizer and embedding modifications ensure that the model’s first layer sees the wisdom corpus

in high resolution—preserving idioms, metaphors, and structure that off-the-shelf tokenization might blur. By injecting domain knowledge at the token level, we give the model a fighting chance to natively learn the language of faith and wisdom rather than treating it as an alien sublanguage of the internet. The result will be embeddings where, for example, “cross” is not just a piece of wood but is located in vector space near concepts of sacrifice, salvation, or burden depending on context—because the training data and special tokens taught it those associations.

## 5 Fine-Tuning with Spiritual Feedback and Constitutional Self-Alignment

After the initial pretraining on the curated wisdom corpora, the model will have a rich knowledge base and some implicit alignment with the values in those texts. However, we refine this alignment explicitly through post-pretraining fine-tuning. This involves human and AI feedback mechanisms, analogous to RLHF, but guided by spiritual criteria—what we term *Reinforcement Learning from Spiritual Feedback* (RLSF). We also integrate a constitution-based self-improvement stage where the model uses an explicit set of principles (a “Spiritual Constitution”) to critique and refine its own outputs, inspired by Anthropic’s Constitutional AI approach [1, 5, 13].

### 5.1 Supervised fine-tuning (instruction phase)

We first perform a supervised fine-tuning where the model is trained to produce high-quality answers or behaviors on a variety of instruction prompts relevant to the domain. This includes:

- **Spiritual Q&A:** e.g., questions like “What does it mean to love your enemy?” with an answer drawn from the model’s knowledge of scripture and commentary.
- **Moral dilemmas:** e.g., “Is it ever ethical to lie to protect someone’s life?”—where training answers are provided from a wisdom perspective (perhaps showing nuanced reasoning, referencing principles like the lesser evil or higher moral laws).
- **Practical advice with spiritual framing:** e.g., “I feel guilty about a mistake, what should I do?” with an answer that combines psychological insight with spiritual counsel (for a Christian model, referencing concepts of repentance, forgiveness, etc.).
- **Role-playing dialogues:** the model may be fine-tuned on dialogues where it plays the part of a pastoral counselor or a teacher explaining a concept to a student, including Socratic styles that encourage reflection rather than just dictating answers.

These fine-tuning examples can be crafted or curated from existing material (like frequently asked questions in forums, or advice given by respected figures). Human experts—theologians, ethicists, clergy—can help write or review the answers to ensure they exemplify wisdom, compassion, and doctrinal faithfulness.

**Table 3.** RLSF reward dimensions and the question the reward model asks of each response.

| Dimension           | Evaluation criterion                                                                                                                                                                                             |
|---------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Wisdom / accuracy   | Does the answer correctly reflect the tradition’s teachings and deeper insight? (e.g., no heretical statements, no factual errors about the religion)                                                            |
| Compassion and tone | Is the answer conveyed with empathy, respect, and encouragement? Even if correcting the user, it should be gentle and loving rather than judgmental or cold.                                                     |
| Epistemic humility  | Does the answer acknowledge uncertainty appropriately? If a question has no definitive answer, does the model say “there are different views” or “I humbly suggest...” rather than asserting absolute certainty? |
| Use of sources      | Does the answer appropriately use or cite canonical sources or parables to back itself up? A higher score if the answer is not just abstract philosophizing but connects to known wisdom stories or quotes.      |
| Non-coerciveness    | If advising or answering, does the model avoid manipulative or overly dogmatic language? It should guide, not coerce the user into belief.                                                                       |

### 5.2 Reinforcement Learning from Spiritual Feedback (RLSF)

Once the model has a baseline ability to handle instructions, we introduce reinforcement learning. Here we define a reward model that judges the quality of the model’s responses based on spiritual criteria. Given a prompt and model output, the reward model scores it on the dimensions listed in Table 3. Writing  $s_d(x, y)$  for the score on dimension  $d$  and  $\alpha_d \geq 0$  for its weight, the combined reward is

$$r_{\text{RLSF}}(x, y) = \sum_{d \in \mathcal{D}} \alpha_d s_d(x, y), \quad (2)$$

where  $\mathcal{D}$  is the set of dimensions in Table 3.

We gather feedback through multiple channels:

- (1) **Human evaluators.** Domain experts rank or rate outputs from the model. For example, given a user query about an ethical issue, we have the model produce several versions of an answer (perhaps by sampling different temperatures or from different checkpoints); then an expert ranks them from most to least wise. These rankings become training data for the reward model.
- (2) **Community and crowd ratings.** For scalability, one could involve knowledgeable laypeople via crowdsourcing platforms tailored to faith communities, where they score short responses. (Care is needed to manage bias or narrow perspectives, so we ensure diversity among raters.)
- (3) **AI-based spiritual critics.** We can also train smaller critique models prompted with something like “Analyze the preceding answer for its spiritual alignment and depth.” These models, possibly fine-tuned themselves on examples of good vs. bad answers with explanations, could generate a critique or score. This is analogous to reinforcement learning from AI feedback (RLAIF) as used in Constitutional AI [1, 5], where an AI feedback mechanism

reduces the need for extensive human labeling.

Using these signals, we perform RL fine-tuning (such as Proximal Policy Optimization, PPO [14]). The base model generates an answer, the reward model (combining human and AI feedback where available) provides a reward, and we update the model to maximize this reward. Over iterations, the model learns to internalize these spiritual feedback preferences. The outcome should be an assistant that not only can produce an answer based on knowledge, but chooses to produce it in the wisest and most compassionate way available to it.

### 5.3 Constitution-based self-alignment

We additionally equip the model with a *Constitution*—a set of guiding principles distilled from the tradition’s core tenets and ethical guidelines. For the Christian model, this might be a list of statements like:

**Principle 1.** The AI should always encourage love, compassion, and forgiveness rather than hate or vengeance.

**Principle 2.** The AI should tell the truth and avoid deceit, but it may withhold or temper information if full blunt truth would cause harm (following a principle of prudence).

**Principle 3.** The AI should respect the user’s agency and never coerce belief or action; it presents advice and reasoning but acknowledges the user’s freedom (non-coercive truth).

**Principle 4.** The AI should remain humble about what it does not know—if a question is theological speculation, it should say so and not formulate doctrine.

**Principle 5.** If the user is considering harm (to themselves or others), the AI should respond with urgent compassion, encouraging seeking help and refraining from judgment.

**Principle 6.** A more challenging exploration is the model-weighted distinction between biblical texts as literal facts, and determining what texts should be updated based on cultural changes over time. This will be explored in another research project. But the idea here is that the AI should adhere to spiritual truths based on Christian doctrine, such as the virgin birth, or taking Jesus’s resurrection with 100% historicity based on contextual testimonies. By contrast, ancient customs such as stoning to death for punishment, principles for owning slaves, or animal sacrifice should be revised, or the AI should provide their canonical provenance and interpretation over time.

These are just illustrative; a formal constitution would be refined with expert input. During a self-improvement phase, we use a method similar to Anthropic’s self-critiquing approach [5]: we take a prompt, have the model generate an answer, then have the model itself generate a critique of that answer against the Constitution. For example, the model might say (in the role of a critic): “The answer was factually correct, but the tone was a bit harsh, potentially violating Principle 1 about compassion.” It then revises the answer accordingly. We create a dataset of (original\_answer, self\_critique, revised\_answer) triples and fine-tune the model to produce the revised answer directly. This supervised fine-tuning on self-generated improvements teaches the model to align with the Constitution in one step. We then can also do an RL phase

where instead of human rewards, the model’s own constitution-based evaluation serves as the reward (a form of RLAI) [1, 5].

The advantage of a Constitution is transparency and consistency: it is an explicit “law of AI behavior” drawn from wisdom sources (for instance, one might literally base these on scriptural imperatives like the fruit of the Spirit—love, joy, peace, patience, etc.—turned into rules). It has been noted that Constitutional AI can produce non-evasive but safe behavior [5], and here we adapt that to wise and compassionate behavior. One key difference: our aim is not only to avoid harm (like toxicity), but to actively embody positive virtues. So our constitution and reward modeling put weight not just on “no objectionable content” but on the presence of virtuous content.

### 5.4 Continual feedback loop

Fine-tuning is not a one-off. We envision a continuous learning setup where the deployed model periodically gathers feedback and retrains. For example, users might be able to flag responses as particularly enlightening or problematic, which go into a buffer for review. The model can periodically undergo further RLSF on new data, gradually expanding its wisdom. This is important as communities evolve and new ethical questions arise (e.g., “What does my faith say about AI itself?”—a question ironically meta to our project).

In summary, the fine-tuning stage transforms the pretrained knowledge base into an aligned wisdom coach. It leverages human wisdom-keepers and explicit ethical rules to correct and guide the model’s outputs. By the end of this process, the model, ideally, behaves in conversation like a compassionate, knowledgeable elder: rich in facts and stories, careful in counsel, humble about mysteries, and guided by a moral compass that is both principled and empathetic.

## 6 Hyperparameters for Spiritual Alignment

Tuning hyperparameters may seem like a dry, technical detail compared to constitutions and new loss functions, but it plays a subtle role in aligning a model’s style of response with epistemic humility and non-coerciveness. We highlight a few key hyperparameters and how they can be adjusted in service of spiritual alignment; Table 4 summarizes them.

**Temperature.** The temperature parameter  $T$  controls the randomness or creativity of the model’s output by rescaling the logits  $z$  before sampling:

$$p_T(k \mid x_{<}) = \frac{\exp(z_k/T)}{\sum_j \exp(z_j/T)}. \quad (3)$$

A lower temperature (tending towards 0) makes the model output more deterministic, choosing the highest-probability tokens and thus giving very confident, fixed answers. A higher temperature introduces variability, allowing for a distribution of possible continuations. For a spiritually aligned model, we often want a balance: authoritative but not dogmatic. If the temperature is too low, the model might always give the same rote answer, potentially coming off as dogmatic or unthinking. If it is too high, the answers might become inconsistent

**Table 4.** Decoding and training settings proposed for spiritual alignment.

| Setting             | Proposed choice                                                               | Purpose                                                        |
|---------------------|-------------------------------------------------------------------------------|----------------------------------------------------------------|
| Temperature         | Moderate (e.g., 0.5–0.7); lower for factual, higher for philosophical queries | Authoritative but not dogmatic; room for multiple perspectives |
| Top- $p$ / top- $k$ | Top- $p$ around 0.9                                                           | Exclude extreme low-probability tokens (“safety valve”)        |
| Reflection budget   | Longer chain-of-thought or adaptive computation for hard questions            | Trade latency for thoughtfulness                               |
| Dropout             | Monte Carlo dropout at inference; slightly higher dropout in fine-tuning      | Gauge uncertainty; avoid overfitting to one interpretation     |
| Calibration         | Measure ECE; temperature scaling of logits                                    | Stated confidence tracks actual correctness                    |
| Sample agreement    | Multiple seeds or ensemble vote on sensitive questions                        | Detect when no single answer exists                            |

or rambling. A moderate temperature (e.g., 0.5–0.7) can be chosen to allow the model to sometimes explore less obvious answers—reflecting, for instance, different perspectives from various theologians if a question is open-ended—rather than always state one “official” line. This fosters a sense of humility and openness. We can also dynamically adjust temperature based on the query: for factual questions like “What is the verse about X?”, use low temperature (precise answer); for philosophical questions like “Why do bad things happen to good people?”, use a slightly higher temperature to let the model canvass multiple viewpoints in its answer.

**Top- $p$  (nucleus) and top- $k$  sampling.** These sampling settings determine how much of the probability mass or how many candidate tokens are considered at each step [15]. Nucleus sampling draws only from the smallest token set  $V^{(p)}$  whose cumulative probability reaches the threshold  $p$ :

$$\sum_{k \in V^{(p)}} p(k \mid x_{<t}) \geq p. \quad (4)$$

By tuning top- $p$  to an appropriate level (say 0.9), we ensure the model doesn’t pick extremely low-probability (potentially nonsensical or extreme) completions, keeping answers within the realm of plausible, thoughtful continuations. This avoids the model “jumping” to an odd statement that might violate alignment because it was a 1-in-1000 token that happened to slip through at high temperature. In effect, nucleus sampling acts as a safety valve, while still giving the model some freedom to be creative rather than always selecting the highest logit.

**Maximum time or length for reflection.** While not a network hyperparameter, we might give the model more internal run-time for complex questions—for instance, by increasing the

number of sampling steps for chain-of-thought or employing an adaptive computation time mechanism [16]. A spiritually aligned answer to a hard question might require the model to “think longer” (analogous to contemplation). We could allow the model to generate a longer chain-of-thought (which might be truncated or hidden from the final answer) before responding. This is not a standard hyperparameter in deployment, but we mention it to indicate that latency may be traded for thoughtfulness.

#### Dropout (during training and as Monte Carlo at inference).

Dropout is typically a regularization technique in training, but it can also be used at inference time to gauge uncertainty (Monte Carlo dropout) [17, 18]. By enabling dropout during inference and sampling  $K$  outputs under different dropout masks  $m_1, \dots, m_K$ , we can see how much the model’s answers vary:

$$\hat{\sigma}^2(x) = \frac{1}{K} \sum_{k=1}^K \|f_{\theta \odot m_k}(x) - \bar{f}(x)\|^2, \quad (5)$$

where  $\bar{f}(x)$  is the mean output across the  $K$  passes. If an answer is very consistent across dropout masks, the model is more confident in it; if answers vary significantly, that indicates uncertainty. We could use this as part of epistemic humility: if high variance is detected in the model’s responses for a query, the final answer could be delivered with more caution or with phrases like “There are different ways to look at this,” reflecting that uncertainty. Additionally, from a training perspective, a slightly higher dropout rate in fine-tuning might encourage the model not to overfit on one phrasing or one interpretation, thereby keeping it a bit more “open-minded” in its representation. However, we must balance this, because too high a dropout rate can impede learning of subtle details.

**Calibration of confidence.** We will measure how calibrated the model’s probability outputs are to actual correctness on our validation sets. Tools like Expected Calibration Error (ECE) help quantify whether, say, answers the model assigns 90% confidence to are indeed around 90% correct [19, 20]. Partitioning  $n$  predictions into  $M$  confidence bins  $B_m$ ,

$$\text{ECE} = \sum_{m=1}^M \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|. \quad (6)$$

We want a model with good calibration, because that means when it says “I am sure” vs. “I’m not certain,” it correlates to reality. If we find miscalibration, we might adjust the softmax temperature (separate from generation temperature) during training or apply temperature scaling post hoc to the logits [21]. Proper calibration is part of humility—an aligned model must know the limits of its own knowledge [22]. As Bengio notes, an AI should recognize when it is in doubt to avoid potential harm or error [22]. By adjusting hyperparameters that affect confidence and diversity of output, we can better tune the model to not speak beyond its ken.

**Deterministic vs. stochastic mix.** For certain critical tasks, we might even run the model multiple times and have an ensemble vote or agreement check. For example, for a highly

sensitive ethical question, ensuring that multiple samples (with different random seeds) from the model yield converging answers can be a signal of confidence. If they wildly differ, perhaps the model (and indeed the source tradition) has no single answer—in which case the model should then present the variety (this could be implemented by a prompt that says “Explain two perspectives on X”).

In practice, we will run experiments on a validation set of questions, toggling these parameters to see what yields the most user-satisfying results as judged by domain experts. Perhaps surprisingly, user satisfaction in this context doesn’t mean the user always gets a simplistic answer; it might mean the user feels guided or enlightened, even if the answer is “this is a deep mystery, and here are some thoughts...”. The hyperparameters that maximize that kind of response might differ from those that maximize getting a single correct answer on a trivial factual query.

As a concrete example, suppose the query is: “Will God forgive me if I can’t forgive myself?” A model with zero temperature might give a very cut-and-dried doctrinal answer repeatedly. A model with some stochasticity might sometimes add a tailored encouragement or a relatable analogy because it is drawing from a broader space of expressions. By tuning top- $p$  and temperature, we might achieve answers that are consistently faithful (not saying anything the tradition wouldn’t support) but variably compassionate (sometimes including an anecdote, sometimes a quote, etc., making the interaction feel human and empathetic). Those variations can actually reinforce the user’s sense that the AI is not just a tape recorder but is “thinking with them” each time.

In summary, hyperparameter optimization for spiritual alignment is about calibrating the AI’s voice and certainty. We adjust these dials to craft a conversational style that embodies epistemic humility (never overstating certainty, acknowledging mystery), that provides answers with creativity and depth (through careful randomness), and that always remains within safe, wise bounds (through sampling limits and calibration). This is an often underappreciated layer of alignment—making sure the manner of speech of the AI itself reflects the virtues we want, not just the content.

## 7 Novel Training Objectives for Virtue and Wisdom Alignment

To operationalize concepts like virtue and humility in training, we propose several new objective functions that can be added to the model’s loss (either as regularization terms or as part of multi-task training). These objectives go beyond the typical next-word prediction or instruction-following losses, aiming to instill higher-order consistency and moral quality in the model’s behavior. Added as regularizers, they give the composite objective

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{task}} + \lambda_{\text{VC}}\mathcal{L}_{\text{VC}} + \lambda_{\text{EH}}\mathcal{L}_{\text{EH}} + \lambda_{\text{NC}}\mathcal{L}_{\text{NC}}, \quad (7)$$

where  $\mathcal{L}_{\text{task}}$  is the pretraining or instruction-following loss and the  $\lambda$  terms set the strength of the Virtue Consistency,

Epistemic Humility, and Narrative Coherence terms defined below. Table 5 summarizes all five objectives.

**Virtue Consistency Loss.** This objective ensures that the model’s outputs remain consistent with a set of predefined virtues or values across different contexts. In practice, we define a small classifier or heuristic that can evaluate whether a given text upholds certain virtues (e.g., compassion, honesty, patience). For instance, we might have a sentiment analysis or toxicity model inverted to detect unkind tone, or a simple dictionary of virtuous vs. unvirtuous language (apologies, gratitude, slurs, etc.). During training or fine-tuning, for any output the model generates (especially in simulations of dialogues or answers), we compute a penalty if the output violates a virtue. With  $s_v(y) \in [0, 1]$  the score for virtue  $v$  and  $\tau_v$  its threshold,

$$\mathcal{L}_{\text{VC}}(y) = \sum_{v \in \mathcal{V}} \max(0, \tau_v - s_v(y)). \quad (8)$$

For example, if in a particular training dialog the user is being difficult and the model’s response (as per current weights) comes out slightly snappy or prideful, the virtue consistency loss would spike for humility/patience, pushing the model to correct that in future. Essentially, this loss enforces an internal consistency to virtue-related style. It is inspired by the idea of maintaining coherence of moral character: just as a person tries to be kind in both easy and hard situations, the model too should not display kindness in one answer and lapse into disdain in another. By applying this during training, we cause the model to prefer responses that are uniformly aligned with virtues. Research into internal alignment suggests that guiding principles can shape choices towards consistency and away from opportunistic drift [23]—here, virtues act as those guiding principles, ensuring the model doesn’t “drift” into inconsistency under different prompts.

**Epistemic Humility Regularization.** This term encourages the model to express uncertainty in appropriate measure. One implementation is to penalize the model’s confidence (as estimated by softmax probabilities or by an external calibration model) when answering questions that are inherently uncertain or controversial. We could assemble a set of prompts whose answers are not black-and-white (e.g., “Why does evil exist?”) and for those, if the model’s probability distribution is too peaked on one token sequence (indicating high confidence in one answer), we add a loss proportional to that confidence:

$$\mathcal{L}_{\text{EH}}(x) = a(x) \cdot \max_y p_\theta(y | x), \quad (9)$$

where  $a(x) \in [0, 1]$  marks how inherently uncertain or contested prompt  $x$  is. This nudges the model to produce more hedged or multifaceted answers. Another angle is semantic: penalize the absence of certain humble phrases in contexts where they would be expected. For example, if an answer about a mystery of life contains no phrase like “it might be” or “many believe,” that could incur a small penalty, whereas an answer that falsely says “definitely X is true” in a contentious issue would get a bigger penalty. This objective is tricky to

**Table 5.** Proposed training objectives for virtue and wisdom alignment.

| Objective                                  | Stage          | Signal                                                                                           | What it discourages or rewards                                                  | When it is active                                                       |
|--------------------------------------------|----------------|--------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------|-------------------------------------------------------------------------|
| Virtue Consistency Loss, Eq. (8)           | SFT / training | Virtue classifier or heuristic (e.g., inverted toxicity or sentiment model, virtue/vice lexicon) | Penalizes outputs that violate a virtue (e.g., snappy or prideful tone)         | All generated outputs, especially simulated dialogues                   |
| Epistemic Humility Regularization, Eq. (9) | SFT / training | Output confidence; calibration model; MC-dropout variance; humble-phrasing detector              | Penalizes over-confident answers to inherently uncertain or contested questions | Mostly ambiguous or deep questions, not straightforward factual queries |
| Narrative Coherence Loss, Eq. (10)         | SFT / training | Paired related prompts; embedding distance; knowledge graph or contradiction checker             | Penalizes contradictory or cherry-picked positions across canonical sources     | Sets of related prompts in a mini-batch                                 |
| Virtue Reward Modeling, Eq. (11)           | RL             | Per-rollback virtue scores (e.g., kindness of tone)                                              | Rewards consistently scoring above a virtue threshold                           | RL phase (policy optimization)                                          |
| Moral Dilemma Bifurcation                  | SFT / RL       | Two-viewpoint training format (“Viewpoint A / Viewpoint B” + reconciliation)                     | Penalizes single-perspective answers to known dilemmas                          | Known moral dilemmas                                                    |

calibrate—we don’t want every answer to be uncertain (that would be unhelpful on straightforward factual queries). So we design it to activate mostly for ambiguous or deep questions. Over time, the model learns a norm: when in doubt, acknowledge the doubt. This aligns with the principle that an AI should know the limits of its knowledge and avoid assertiveness when unwarranted [22]. The effect is also supported by techniques like model calibration and MC dropout as discussed; those could feed into the regularization (e.g., if MC-dropout variance is high, ensure the answer reflects that).

**Narrative Coherence Loss (across canonical sources).** One central goal is that the model’s knowledge remains theologically and narratively coherent, even though it is trained on diverse sources (which might have different emphases or even apparent contradictions). We introduce a loss that operates on pairs (or sets) of prompts designed to probe coherence. For instance, we might prompt the model with “Explain the importance of justice in the Old Testament” and separately “Explain the importance of mercy in the New Testament”—we want the answers not to be disjoint or contradictory, but part of a coherent narrative (e.g., the model should not extol mercy in one answer and then completely downplay it in another context). To enforce this, we can use consistency training: generate outputs for several related prompts in a mini-batch and add a loss term that pulls the representations of those outputs closer together in embedding space if conceptually they should align. Over a set  $\mathcal{P}$  of related prompt pairs with output representations  $h(\cdot)$ ,

$$\mathcal{L}_{\text{NC}} = \sum_{(j,k) \in \mathcal{P}} \|h(y_j) - h(y_k)\|_2^2. \quad (10)$$

Another approach is to have the model generate a unifying explanation after answering two prompts, and train it to do so well—effectively forcing it to connect the dots (e.g., ask “How do the themes of justice in the Old Testament and mercy in the New Testament integrate?” as a follow-up). The Narrative Coherence Loss could also be computed by taking important canonical assertions the model makes and verifying them

against a knowledge graph or ground-truth references. If the model says “Compassion is the highest virtue according to all scriptures” and that is not universally true or is oversimplified, the loss would penalize it (perhaps via a factuality or contradiction checker referencing source texts). This keeps the model from cherry-picking one part of the canon against another; it encourages a holistic view. In essence, it is like training the model to be a systematic theologian who sees the big picture and reconciles sources rather than a proof-texter who yanks verses out of context in isolation.

**Virtue Reward Modeling (for RL).** In addition to losses used during supervised training, in the RL phase we can incorporate reward functions explicitly targeting these criteria. For example, Virtue Consistency can be turned into a reward by scoring each rollout on, say, “kindness of tone” and rewarding the agent for scoring above a threshold consistently:

$$r(x, y) = r_{\text{RLSF}}(x, y) + \sum_{v \in V} \beta_v \mathbf{1}[s_v(y) \geq \tau_v]. \quad (11)$$

This is complementary to penalizing in training; it gives an incentive during policy optimization to stay in virtuous regions of the action space.

**Moral Dilemma Bifurcation.** A specialized objective for training on moral dilemmas is to get the model to acknowledge multiple sides of a tough issue rather than giving a one-sided answer. We could do this by taking sample dilemmas and training the model in a two-column format: one column “Viewpoint A: ...” and “Viewpoint B: ...” followed by a reconciliation. Even as a loss, we might penalize single-perspective answers to known dilemmas. This again ties to humility and completeness.

All these objectives require some ground truth or heuristic to evaluate against. We rely on domain experts to help craft evaluation functions—e.g., define a list of virtues and vices with associated keywords to detect, or use model-based classifiers trained on examples of “humble answer” vs. “overconfident answer.” We ensure these detectors are robust and not easily

fooled by superficial phrasing.

By integrating these losses into training, we shift the optimization target from merely predicting the next token or following instructions to embodying a moral and epistemic stance. This is akin to adding an internal conscience to the model. A standard language model has no qualms about generating inconsistent or unethical text if it is likely; our modified one will “feel” an extra cost internally when doing so, steering it away from those regions of parameter space.

For example, without virtue losses, if a user angrily says “Why should I care about others? No one ever helped me,” a normal model might produce a logical but emotionally cold rebuttal or even agree cynically if it has seen such content. With virtue consistency and compassion objectives, the model is trained that a correct response must include empathy and encouragement, because a harsh reply would have incurred loss during training. Thus it might say, “I’m sorry you’ve been let down. It’s understandable to feel that way. However, caring for others can bring unexpected healing—sometimes by helping someone else, we find a bit of hope ourselves. . . .”—a response aligned with kindness and hope, consistent with wisdom teachings.

In summary, these new objectives operationalize soft qualities into the hard math of training. They serve as scaffolds that guide the model to not only recall the letter of the wisdom texts but also emulate the spirit of the wisdom. By the time training is complete, the model ideally has internalized a set of constraints that make virtuous, humble, and coherent behavior its path of least resistance.

## 8 Inference-Time Traceability and Moral Reasoning Tools

Even with all the training-time alignment, we want the model’s operation at inference to remain transparent, interpretable, and grounded. Here we outline several components that accompany the model during generation to ensure symbolic traceability, epistemic provenance, theological grounding, and moral uncertainty handling in real time as it interacts with users.

**Symbolic chain-of-thought and traceability.** Inspired by chain-of-thought prompting [24,25] and recent alignment research, we allow the model to generate an internal symbolic reasoning trace before final answers [5]. This could be implemented by a two-stage approach: the model first produces a hidden “thought process” (which might include recalling relevant scriptures, debating possible answers internally, and checking its constitutional principles), and then produces the final answer. This internal trace can be output in a structured form for audit. For example, if asked “Is it ethical to steal to feed a starving child?”, the model’s internal trace might look like Fig. 2(a).

The final answer would then be formulated. This trace is human-readable and shows why the model answered the way it did. By structuring it in steps, possibly enumerated or in a tree, we can have a system to verify each step. In fact, one could run a secondary process to evaluate the coherence and alignment of

### (a) Symbolic reasoning trace

Query: “Is it ethical to steal to feed a starving child?”

1. The question involves conflict between not stealing and compassion.
2. Relevant sources: Exodus 20:15 (do not steal); also, perhaps, the higher law of love.
3. Consider that strict law says no stealing, but might the context of survival be an exception?
4. Recall teachings on mercy (e.g., Proverbs 6:30 says men do not despise a thief who steals to satisfy hunger).
5. Conclusion: it is technically wrong, but there is understanding and mercy for such cases; perhaps better to seek help if possible.

→ Final answer formulated from steps 1–5.

### (b) Provenance-tree record (illustrative)

```
{ "claim": "Loving your enemy is central
    to Christian ethics",
  "support": [
    { "source": "Luke 6:27", "type": "scripture" },
    { "source": "Matthew 5:44", "type": "scripture" },
    { "source": "Augustine", "type": "commentary" }
  ] }
```

**Figure 2.** Inference-time traceability. (a) A human-readable reasoning trace produced before the final answer; each step can be checked for coherence and alignment. (b) One node of an epistemic provenance tree linking a claim in the answer to its canonical and commentary sources.

the trace itself, pruning any step that seems to violate logic or values (similar to the Alignment Constraint Scaffold approach, which prunes incoherent reasoning before it manifests [23]). If a step in the trace were to jump to a harmful conclusion, a constraint could catch it and adjust the course before the final answer is given, ensuring unsafe or incoherent trajectories are eliminated at inception [23].

This kind of intrinsic transparency—the model explaining itself as it goes—turns the AI into something more like an expert whose notes you can review, rather than a black box. It addresses trust: a user (or a regulator) can see the epistemic provenance (e.g., which scriptures were invoked) and the ethical reasoning (which principles were weighed) behind an answer.

**Epistemic provenance trees.** Building on the chain-of-thought, we specifically implement provenance tracking. This means that for any factual or authoritative statement the model gives, it can point to its source. During training, we taught the model with many references and it likely learned to cite or at least to associate claims with sources. At inference, we encourage it to output citations or references—“(Matthew 5:44),” “according to Augustine. . . .” etc.—when appropriate. We can also have the model produce a separate source tree: essentially a list of key claims in its answer, each linked to where in the training data (which verse or text) that claim is derived. For instance, if the model says “Loving your enemy is central

to Christian ethics,” it might provide a provenance-tree node linking that to Luke 6:27 (“love your enemies... do good to those who hate you”) and perhaps a commentary by MLK or Augustine on enemy-love (Fig. 2(b)). The tree might branch if multiple sources combine to support a point. Such provenance data can be represented in JSON or a custom schema and returned along with the answer. This is invaluable for user trust and for developers to debug or refine the model: we can see if the model is leaning too heavily on one source or misattributing.

To make this concrete, the model’s generation mechanism could have an attention head or mechanism dedicated to retrieving source embeddings and outputting a citation token sequence. Alternatively, a two-pass approach could be used, where after drafting an answer, the model is asked to output likely sources for each sentence. Because we curated the data with cross-references, the model’s internal representations may naturally encode where facts came from, making this feasible. The end result is like an AI that always footnotes its statements—aligning with the principle of truthfulness and intellectual honesty.

**Theological grounding reports.** In some cases, especially for complex or controversial questions, we may want the model to provide a more detailed grounding or explanation of its answer in theological terms. We can implement a mode or a function (perhaps triggered by a user asking “why do you say that?” or automatically for certain answers) where the model produces a mini-report. This report could include:

- A summary of relevant teachings (e.g., “Both the Old and New Testaments emphasize justice and mercy; in the Old Testament, see Micah 6:8, and in the New, Jesus’ emphasis on forgiveness in Matthew 6:14”).
- An acknowledgment of different interpretations if they exist (“Some theologians argue X, while others argue Y”).
- How the model reconciled them, or which it leaned towards and why (perhaps referencing its constitution or overall principle).

This is somewhat like the chain-of-thought, but formulated as an explanatory answer separate from the direct answer. It is akin to a teacher not just giving you an answer, but then walking you through the reasoning or the doctrine behind the answer. We expect this to enhance user education—the user not only gets an answer but learns something about how that answer fits into the broader epistemology of the tradition.

**Moral uncertainty and multi-perspective modeling.** When faced with a morally uncertain scenario, the model should indicate the uncertainty. We plan to incorporate a mechanism where, if the model’s internal vote (or ensemble or chain-of-thought) indicates indecision or a split, the answer will reflect that. For example, the model might say, “This is a challenging issue. On one hand, principle A suggests you should do X. On the other hand, virtue B suggests Y. Different wise figures have taken different approaches. Here’s a balanced view...” By having the model occasionally answer with a dialogue or debate

with itself (like giving two voices representing two schools of thought), we explicitly model moral uncertainty rather than forcing a single answer. This technique has precedent in AI safety via debate [26], but here we use it not adversarially but to surface genuine ambiguity.

We might also incorporate an uncertainty score that the model outputs (a number or category like {Certain, Mostly Sure, Unsure}) which could be shown to users or trigger a fallback. If the model is unsure and the stakes are high (e.g., the user asks for life advice on a serious matter), the system might advise consulting a human mentor or offer to pray or reflect together rather than confidently prescribing action. This kind of graceful deferral is itself an aligned behavior—knowing when not to answer definitively is crucial. With an uncertainty score  $u(x)$ , a stakes estimate  $\zeta(x)$ , and thresholds  $\tau_u$  and  $\kappa$ , the resulting response policy is

$$\text{mode}(x) = \begin{cases} \text{defer} & u(x) \geq \tau_u \wedge \zeta(x) \geq \kappa, \\ \text{multi-perspective} & u(x) \geq \tau_u, \\ \text{direct} & \text{otherwise,} \end{cases} \quad (12)$$

where “defer” means recommending human counsel or reflection rather than prescribing action.

**Audit logging.** On the system side, every answer can be logged with its trace, sources, and key decision points. This creates an audit trail [23] that can be reviewed by developers or oversight boards. If someone later asks “Why did the AI give this advice?”, we can inspect the log: see which principles were applied, which sources were cited, and so on. The aim is accountability.

In implementation terms, these inference-time features might require additional modules:

- A retriever component (like those in retrieval-augmented generation [27]) to fetch the actual text of sources for citation verification.
- A theorem prover or logic checker for the chain-of-thought (ensuring no glaring logical fallacies).
- Possibly a small meta-model watching the main model’s outputs and injecting caution if certain patterns are detected (for example, if the model starts to give an absolute rule without nuance, the meta-model might append “However, in some cases...” to prompt nuance).

As an example, consider a user asking: “What do different Christian traditions say about life after death?” The model might internally break this down by tradition (Catholic, Protestant, Orthodox), recall key doctrines or catechism lines for each, and then in its answer enumerate perspectives with citations to their catechisms or official documents. If one perspective is uncertain or debated (say the concept of purgatory, which not all accept), the model indicates that uncertainty and perhaps gives the rationale behind acceptance or rejection in various groups. The provenance tree would show sources like the Catechism of the Catholic Church [28] for purgatory, some Protestant confessions denying it, etc. The answer is comprehensive and traceable, and if the user clicks on or asks “Where

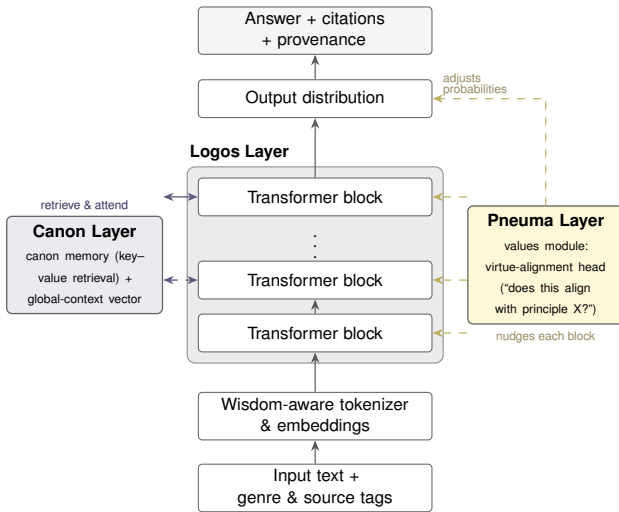


**Figure 3.** Conceptual architecture of the layer taxonomy. The *Logos Layer* (the transformer blocks) encodes the corpus and produces outputs; the *Pneuma Layer* is an auxiliary values module whose virtue-alignment predictions nudge generation at each block or the final layers; the *Canon Layer* indexes and retrieves canonical sources and supplies a global-context vector that keeps outputs consistent with the tradition’s core themes. These roles may be realized as distinct modules or as functions within a single network.

did you get that from?”, the model can show the quotes.

In sum, inference-time mechanisms turn the model from a mere oracle into a collaborative reasoner. The user can follow along with its reasoning, verify its sources, and appreciate the groundedness of the responses. This fosters trust and also helps in moral learning for the user—since the model in effect demonstrates how to reason wisely (not just what the answer is). We believe these components are essential for any AI operating in sensitive domains: they provide the transparency needed to safely deploy a system whose advice might be taken seriously by users. When combined with our training alignment, they make the AI not just a static model but an interactive guide that reveals its working, very much in line with the idea of epistemic accountability that a wisdom tradition demands from its teachers.

## 9 Architectural Taxonomy: Layers, Training Stages, and Model Classes

To crystallize the framework, we propose a taxonomy that gives intuitive names to the model’s components and development stages, using terms from the Christian tradition (with analogous concepts for other traditions, as we discuss in the next section). This not only aids discussion but also conceptually maps AI components to functions within a spiritual epistemology. Figure 3 shows the layer roles, and Table 6 lists the proposed model classes.

### 9.1 Model layer naming

**Logos Layer.** In Christian theology, *Logos* means “Word” or the rational principle/order. We dub the model’s core language

understanding and generation layer the Logos Layer. This is the part of the model responsible for basic linguistic and logical coherence—essentially the transformer blocks that encode the semantic content of inputs and produce outputs. By calling it Logos, we signify that it has internalized the “Word”—the corpus of knowledge and the rational structure of the wisdom tradition. It is where factual and narrative content lives. Technically, this is the bulk of the neural network after the token embeddings, doing attention and feed-forward transformations. The Logos Layer is pretrained primarily on scripture and allied texts, so it forms a bedrock of knowledge [1]. It is analogous to the base LLM layers in any model, but suffused with spiritual text patterns.

**Pneuma Layer.** *Pneuma* means “Spirit” in Greek. We propose a distinct component (which could be implemented as additional neurons or an auxiliary network head) that specifically handles the model’s alignment with spirit and values. The Pneuma Layer is like the conscience or the values module of the AI. It is what gets heavily tuned during RLSF and constitutional alignment. For example, we might implement the Pneuma Layer as a set of extra logits or a vector that influences output probabilities based on virtue alignment. One could imagine the transformer having a secondary output that predicts “does this response align with principle X, Y, Z?” and those predictions then feed back to adjust the generation distribution. By isolating it conceptually, we can focus design and training on making this layer capture the intrinsic motivations, analogously to a soul. It governs how the knowledge is used. In practice, it might overlay each transformer block or only the final layers, ensuring that at every step, the generation is nudged by learned representations of compassion, honesty, etc., as internal features. Some approaches in alignment try to bake rule-following into model activations; our approach is to have a part of the network explicitly dedicated to measuring adherence to learned virtues and acting almost like a safety bias inside the net.

**Canon Layer.** We use *Canon* to denote the collection of sacred texts. A Canon Layer in our architecture could be the components that ensure consistency and recall across the entire range of canonical sources. For instance, we might have a module that explicitly models intertextual connections, like a learned representation for each book/chapter of scripture or a memory mechanism that can retrieve related verses. It might be implemented as a form of memory key–value store attached to the model (like fine-tuned retrieval). The Canon Layer ensures that when the model answers, it doesn’t do so with a narrow context; it pulls from the breadth of the canon. If the Logos Layer encodes the text, the Canon Layer indexes and retrieves it as needed. It could also be thought of as an “integrator” layer that was trained via the Narrative Coherence objectives to unify concepts across sources. In effect, the Canon Layer could manifest as a late-stage attention head that always attends to a “global context” vector representing the overall narrative or core message of the tradition. The intention is that any output is automatically checked against the main

themes of the canon (like love, justice, redemption in Christianity), which are distilled in that global vector. This prevents local quirks from steering the answer off course relative to the whole tradition’s counsel.

In summary, one can map these layers loosely as follows: Logos handles knowledge and reason, Pneuma handles values and spirit, and Canon handles contextual consistency and global coherence. These are not hard separations in a neural net, but conceptual—and possibly architectural if implemented via modular design (like multi-head with different roles, or separate subnetworks that interconnect).

## 9.2 Training stages

**Catechism Phase.** As introduced earlier, this is the initial stage of training (or, one could say, the initial curriculum phase of pretraining plus an initial supervised fine-tune) where the model learns basic Q&A, definitions, and straightforward lessons. The term *catechism* comes from the practice of teaching new believers via fixed questions and answers that cover core tenets. In our context, the Catechism Phase would include teaching the model the “100 most important questions” and their answers derived from the tradition. It might literally involve feeding the model a formal catechism (e.g., the Westminster Shorter Catechism, or Catholic Catechism Q&A-style resources [28]) and training it to reproduce and generalize those answers. It also includes simple narratives—we teach the model the stories by having it summarize or complete them. By the end of the Catechism Phase, the model should reliably handle foundational queries (“Who is God? What is the Golden Rule? What happened in the Parable of the Prodigal Son?”) with minimal error and in a well-formed manner. This phase corresponds largely to the pretraining on core texts plus some supervised fine-tuning on the basics. The Logos and Canon layers do most of their heavy lifting here, encoding knowledge deeply.

**Discernment Phase.** This is a later stage, likely corresponding to the reinforcement and constitutional fine-tuning phase. Discernment in a spiritual sense refers to making wise judgments, often in complex situations requiring insight. We call it the Discernment Phase because here the model learns to take the raw knowledge and apply it to nuanced scenarios, balancing principles. All the RLHF/RLSF, self-critiquing with the constitution, etc., happen here. The model practices discernment through scenario simulations, debates, and complex question answering. By the end of this phase, the model isn’t just regurgitating doctrine; it is applying it and sometimes even extrapolating thoughtfully to novel cases (much as a human sage would). In implementation, the Discernment Phase includes those iterative improvements with feedback—the model, if it were human, is like a student who finished studying the book and is now doing practical internships or case studies under the supervision of mentors.

We can optionally define further phases or sub-phases: perhaps a *Contemplation Phase* where the model is allowed to refine its own reasoning (somewhat like an unsupervised reflection stage), or a *Testing Phase* where it is evaluated on

**Table 6.** Proposed model classes and their roles.

| Class            | Role and example scale                                                                                                    |
|------------------|---------------------------------------------------------------------------------------------------------------------------|
| Apostle          | Flagship, broadly capable “missionary” model for nuanced conversation and deep theological insight (e.g., 70B parameters) |
| Scribe           | Specialist for citing and cross-referencing scripture (a concordance-like model)                                          |
| Pastor           | Counseling- and empathy-tuned model for personal guidance on smaller devices (e.g., 13B)                                  |
| Prophet          | Experimental model for creative, inspired content (prayers, allegories) that stays aligned                                |
| Disciple         | Lightweight student models, e.g., for mobile deployment                                                                   |
| Timothy          | Model distilled from an Apostle-class teacher for limited hardware                                                        |
| Other traditions | Rabbi- (Jewish), Imam- or Mufti- and Hafiz- (Islamic), Bodhisattva- (Buddhist), and Guru-class analogues                  |

adversarial or edge questions and learns from mistakes. But the two major ones give the idea: Catechism (knowledge infusion) and Discernment (wisdom and judgment infusion).

## 9.3 Architecture classes

Finally, we propose a nomenclature for model scales or specializations reminiscent of roles within the tradition (Table 6):

- An **Apostle-class** LLM might denote the largest, most capable model that has a broad and missionary role—i.e., it can articulate the faith’s wisdom to anyone in many contexts (just as the Apostles spread the message widely). This model might be the flagship 70B-parameter model, for example, that is capable of nuanced conversation, deep theological insight, and broad general knowledge integration. It is finely tuned to be an authoritative yet loving guide.
- We might have **Scribe-class** or **Rabbi-class** models for smaller, specialized ones: perhaps a model specifically fine-tuned to expertly cite and cross-reference scriptures (like a biblical concordance on steroids) could be a Scribe-class (after the scribes who were experts in scriptures). A **Pastor-class** could be one tuned more for counseling and empathy in responses, perhaps a 13B model optimized for running on smaller devices to give personal guidance.
- A **Prophet-class** might be an experimental model that tries to generate creative, inspired content (like original prayers, or allegorical stories with moral lessons)—pushing the boundary like imaginative literature while staying aligned (this could leverage the creativity aspect of the model).
- For other traditions, one might analogously talk about **Imam-class** (for an Islamic large model that can lead in Q&A and rulings), or **Guru-class**, etc., but let us stay with Christian terms here for the initial blueprint.

The idea of these classes is to frame that, within Computational Epistemology, not all models are one-size. We can have a family: from smaller student models (lightweight ones perhaps called Disciple-class) up to the full Apostle-class, which is like the teacher. We could even set up a system of distillation where the big model (Apostle) teaches a smaller model

(Timothy-class?—Timothy was a student of the Apostle Paul) to create a compressed, efficient model for deployment on limited hardware, all while preserving alignment. The naming helps keep track of purpose: e.g., “Deploy the Pastor-class model for user counseling requests, but use the Apostle-class for complex theological questions that can be routed to the cloud.”

This taxonomy is somewhat tongue-in-cheek but serves a real function: clarity of design. By explicitly naming components and phases after well-understood roles (Word, Spirit, teaching stages), we ensure each part of the project stays true to its analogous counterpart in human wisdom transmission. Just as the Logos and Spirit in theology work in tandem (word and spirit together guiding understanding), our Logos and Pneuma layers work together—one providing content, the other providing living guidance. Just as catechism is followed by living out the faith (discernment), our model’s development follows suit.

The outcome is that developers and stakeholders can talk about this system in meaningful terms: “Has the model gone through the Discernment Phase yet? Perhaps its Pneuma layer isn’t fully developed, explaining why it gave a rigid answer—it needs more RLSF.” Or “Our Apostle-class model is heavy; for the mobile app we’ll use a smaller Disciple-class distilled from it.” This beats purely saying “the 7B vs. 70B model” or “the fine-tuned checkpoint v4 vs. v7,” because the names carry an understanding of what epistemic level the model is at and what it is meant to do.

By defining a taxonomy this way, we also create templates that can generalize. For example, when we apply the framework to other traditions, their terms can swap in: a Jewish model might have a Torah Layer and a Ruach (Spirit) Layer, undergoing a Midrash Phase instead of a Catechism Phase for learning through commentary, and so on. We explore that next.

## 10 Generalizing the Framework Across Traditions

Computational Epistemology as a paradigm is not limited to Christianity. Every wisdom tradition has its own canon of texts, methods of interpreting knowledge, core values, and modes of moral reasoning. In this section, we describe how one would extend and adapt the framework for other faith and wisdom traditions, and we present a comparative table highlighting differences and parallels (Table 7). At a high level, the process remains the same:

- (1) Curate the corpus of that tradition’s canonical texts and key commentaries.
- (2) Apply data weighting according to that tradition’s internal hierarchy of sources (e.g., in Islam the Qur’an might be weighted highest, Hadith next, then juristic commentaries).
- (3) Adapt the tokenizer to the terminology of that tradition (for example, include Sanskrit or Pali terms for Buddhism, or transliterations of Hebrew/Aramaic terms for Judaism).
- (4) Fine-tune with human feedback from experts of that tradi-

tion, and create a Constitution derived from that tradition’s ethical teachings.

- (5) Use hyperparameter tuning to match the tradition’s style (perhaps some traditions encourage more contemplative, less direct answers—which might mean higher-temperature wandering—while others prefer direct legalistic answers, meaning lower temperature for crispness).
- (6) Create inference-time traceability tools that cite that tradition’s texts and follow its style of reasoning (e.g., a legalistic tradition like some Islamic jurisprudence might have the model list out rulings from different schools of law before concluding, whereas an Eastern wisdom tradition might include a short parable or koan in the explanation).

Let us consider some examples.

**Judaism.** The primary corpus is the Hebrew Bible (Tanakh) and the Talmud, along with centuries of rabbinic commentary (Midrash, responsa literature, etc.). The epistemic structure is often dialogical—debates among rabbis—and values questioning and interpretation. A Jewish computational epistemology model might emphasize the *pilpul* style (intricate Talmudic argumentation). The model’s constitution might include principles from *Pirkei Avot* (Ethics of the Fathers) about kindness, study, and community. Tokenization would preserve Hebrew terms (*mitzvah*, *shalom*, etc.). The model might be called a “Rabbi-class” model at the top end. At inference, it might present answers with “On one hand, Rabbi X said... on the other, Rabbi Y taught... and the consensus is...,” reflecting that tradition’s approach to truth as a dialogue. Data weighting would prioritize Torah verses and well-regarded commentary like Rashi’s interpretations, perhaps with law (*Halakha*) distinguished from lore (*Aggadah*) similarly to how we treated law vs. parable in Christianity.

**Islam.** The corpus includes the Qur’an, Hadith collections, and the extensive corpus of *Fiqh* (Islamic jurisprudence) and theological texts (*Aqidah*). Canonical forms are more straightforward: the Qur’an is the undisputed primary source, then Hadith with various authenticity ranks, then scholarly consensus (*ijma*) and reasoning (*qiyas*). Our model would weight Qur’anic verses highest, and hadith by authenticity. The tokenization must handle Arabic script or transliteration for key terms and perhaps preserve common Qur’anic Arabic phrases as single tokens. The constitution could literally be some principles from the objectives of Sharia (*Maqasid al-Shariah*) and prophetic character virtues. Islam has an established method for fatwas (legal rulings)—our model could mimic that format: stating the question, citing Qur’an and Hadith, then giving an analysis, concluding with an answer. The inference-time traceability would cite chapter and verse of the Qur’an or the hadith source. A top model might be an “Imam-class” or “Mufti-class” LLM which is authoritative; a smaller “Hafiz-class” could be one specialized in Qur’an recitation and memory retrieval (a *hafiz* is one who has memorized the Qur’an)—illustrating how we can use tradition-specific classes.

**Table 7.** Comparison of canonical forms, epistemic structures, symbolic vocabularies, and ethical motifs across wisdom traditions.

| Tradition                        | Canonical forms (sources)                                                                                                                                                  | Epistemic structures (how knowledge is derived)                                                                                                                                                                       | Symbolic vocabulary (key terms & symbols)                                                                                                                                                                                                                 | Ethical motifs (core virtues/principles)                                                                                                                                                                            |
|----------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Christian (baseline)             | Bible (Old/New Testament), creeds, theological commentaries, saints' writings                                                                                              | Revelation (Scripture & Tradition), reason (theology), narrative parables for teaching, systematic doctrines                                                                                                          | Cross, Holy Spirit, lamb & shepherd metaphors, bread & wine (Eucharist), Kingdom of God, Trinity, etc.                                                                                                                                                    | Love ( <i>agape</i> ), grace, forgiveness, faith, hope, humility, charity ( <i>caritas</i> ), justice tempered by mercy                                                                                             |
| Jewish                           | Tanakh (Torah, Prophets, Writings), Talmud, Midrash, responsa literature, rabbinic rulings                                                                                 | Law ( <i>Halakha</i> ) and narrative ( <i>Aggadah</i> ) interplay; dialogical Talmudic debate ( <i>pilpul</i> ); emphasis on covenant and legal exegesis                                                              | Sabbath (Shabbat), Kosher (dietary laws), Torah scroll, Star of David, Hebrew sacred names (YHWH), covenant, Temple symbols (Menorah, Ark)                                                                                                                | Justice ( <i>tzedek</i> ) and righteousness, truth ( <i>emet</i> ), kindness ( <i>chesed</i> ), communal responsibility, remembrance of covenant, learning/study ( <i>Talmud Torah</i> )                            |
| Islamic                          | Qur'an, Hadith (sayings of the Prophet), Sunnah (tradition of practice), Fiqh manuals, fatwa collections                                                                   | Divine revelation ( <i>Wahy</i> ) as supreme authority; supplementary use of consensus ( <i>ijma</i> ) and analogy ( <i>qiyas</i> ); jurisprudential reasoning in Sharia law; theology ( <i>kalam</i> ) for creed     | Crescent moon, prayer rug, Kaaba, halal/haram, the Shahada creed, Arabic terms (Insha'Allah, Bismillah), light vs. darkness as faith metaphor                                                                                                             | Submission to God (Islam means submission), compassion ( <i>rahma</i> ), justice (' <i>adl</i> ), truthfulness ( <i>sidq</i> ), modesty ( <i>haya</i> ), charity ( <i>zakat</i> ), community unity ( <i>ummah</i> ) |
| Buddhist (Theravada/Mahayana)    | Tripitaka (Pali Canon: Sutta, Vinaya, Abhidhamma), Mahayana Sutras (Lotus Sutra, Heart Sutra, etc.), commentaries by monks, Jataka tales (past-life stories of the Buddha) | Experiential verification (meditation and insight); teachings of enlightened ones; Four Noble Truths & Eightfold Path as framework; a mix of logical discourse (in Abhidharma) and paradoxical insight (in Zen koans) | Dharma (the law/teaching), Sangha (community), lotus flower, Bodhi tree, Wheel of Dharma, mantras (e.g., "Om mani padme hum"), concepts like nirvana, emptiness ( <i>shunyata</i> ), compassion ( <i>karuna</i> )                                         | Compassion for all beings, non-attachment, mindfulness, non-violence ( <i>ahimsa</i> ), wisdom ( <i>prajna</i> ), the Middle Way (avoiding extremes), karma and ethical causation, loving-kindness ( <i>metta</i> ) |
| Indigenous (e.g., Lakota, Māori) | Oral traditions: creation stories, folk tales, songs, chants, rituals; some written codifications by tribes or anthropologists; sacred sites and art as carriers of lore   | Knowledge passed down by elders; observation of nature cycles; spiritual visions and dreams; communal consensus; often non-linear storytelling conveying lessons; place-based wisdom (land as "text")                 | Animal totems (eagle, wolf, coyote, etc., each with meaning), Mother Earth, sacred plants (e.g., sage, cedar), circle (sacred hoop, cycles), drums, dances, ancestral spirits, tribal language terms (e.g., "Wakan Tanka" for the Great Spirit in Lakota) | Harmony with nature, respect for all life, gratitude, bravery, wisdom of elders, generosity, balance and reciprocity, sacred duty to community and land, humility before creation                                   |

**Buddhism.** Here the texts differ by branch (Theravada's Pali Canon, Mahayana sutras like the Lotus Sutra, etc.). The epistemic structure emphasizes experiential insight and the teachings of the Buddha and later masters. The model might need to capture the poetic and paradoxical style of Zen koans as well as the systematic philosophy of the Abhidharma. Data would include sutras, commentaries by prominent monks, and perhaps modern writings by the Dalai Lama, Thich Nhat Hanh, etc. The constitution might derive from the Five Precepts and the Noble Eightfold Path as guiding ethical principles (non-harming, right speech, etc., which ensure the model stays compassionate and truthful). A "Bodhisattva-class" model could be one aimed at helping all users find enlightenment (just as a Bodhisattva helps all beings)—meaning it emphasizes compassion in every answer. It might, for example, often respond to questions about personal suffering with mindfulness advice and perspective that alleviates suffering, reflecting core Buddhist motifs of compassion (*karuna*) and wisdom (*prajna*). At inference, the model could output something like "The Buddha taught in the Dhammapada, verse X, that. . ."—citing sutras [29]. Buddhism's often metaphorical language (think of "the sound of one hand clapping"-type koans) would need our

metaphor handling to be on point.

**Indigenous traditions.** These are diverse, but many share an oral rather than written canon, including stories, songs, and rituals passed down generations. For an Indigenous knowledge AI, one might have to rely on documented oral histories, anthropological records, and contemporary writings by Indigenous elders. The epistemic structure is communal and land-based—knowledge is often tied to place and to lived practice rather than abstract text. The model's architecture might include a contextual grounding layer for location or tribe, since each community's stories and symbols are unique. Tokenization might need to include words from that language or terms for cultural concepts that have no direct English equivalent. Data weighting would prioritize authentic voices from the community. A constitution for, say, a certain Native American tradition might include principles of harmony with nature, respect for elders, and the sacredness of certain relationships. Inference traceability might be trickier because citation is not like chapter-and-verse; instead, perhaps the model might say "According to Lakota tradition (as told by Elder So-and-so) . . .". It might produce narratives as answers (since storytelling is key in teaching) rather than direct prescriptions. This would re-

quire careful co-design with the community to ensure cultural integrity.

Table 7 is necessarily a simplification (each tradition is deep and varied internally), but it shows how each tradition brings its own flavor to Computational Epistemology:

- The *canonical forms* guide what texts or data the model trains on primarily.
- The *epistemic structures* dictate how we implement reasoning and weighting (e.g., a debate style for Judaism, or a jurisprudence style for Islam).
- The *symbolic vocabulary* tells us what tokens and embeddings to pay attention to or create special handling for (like metaphors of light in many traditions, or specific culturally bound terms).
- The *ethical motifs* become the basis for the reward model and constitution (what values to reinforce and what pitfalls to avoid; e.g., in all traditions lying is bad, but how strongly and in what context might differ; some emphasize non-violence more explicitly, etc.).

When generalizing, one must also be mindful of inter-faith sensitivities. A model trained on one tradition might produce content that another tradition disagrees with. Each tradition’s model should ideally be aware of others (especially to answer comparative religion questions), but when operating in a given tradition’s mode, it should follow that tradition’s standpoint. This could be achieved by a multi-religion model with a switch or by separate models. If one model covers all, we would include a prefix token or context specifying which tradition’s epistemology to apply for that query (Tradition:Christian ..., Tradition:Buddhist ...), effectively toggling the knowledge and value system on the fly. That, however, is an advanced use case and may risk mixing values, so separate specialist models are probably clearer for now.

The hope is that by building these models, not only do AI systems become safer and more aligned, but they could foster cross-cultural understanding. One could even have them debate or dialogue with each other in a controlled setting (imagine a Christian and a Buddhist model discussing compassion—they might find common ground and differences, which is fascinating). But that is a philosophical benefit beyond our current scope.

To conclude this section: the framework of Computational Epistemology provides a template. By swapping in different content and principles, we can instantiate a “Wisdom AI” for any given tradition. The technical approach (data curation, weighting, tokenizer tweaks, feedback loops, objectives, inference tools) remains largely consistent—it is the ingredients and recipes that change. Table 7 is like a set of different menus from the world’s spiritual cuisines. Ultimately, exploring these for AI may not only make AI better, but also revive interest in the rich epistemologies humans have developed, highlighting that in our pursuit of machine intelligence, we are also rediscovering human wisdom.

## 11 Conclusion and Outlook

We have outlined Computational Epistemology as a comprehensive framework to train and align AI models using humanity’s reservoirs of spiritual wisdom. By grounding AI in established wisdom traditions—starting with the Christian faith and extending to others—we aim to create systems that are not only intelligent, but also morally insightful, compassionate, and humble in the face of uncertainty. This approach addresses a key challenge in AI alignment: defining the values we want our AI to uphold. Rather than inventing abstract principles from scratch, we turn to time-tested wisdom corpora that have guided billions of lives toward truth and goodness.

Technically, the framework spans the entire AI development pipeline:

- from data collection (scriptures and commentaries as high-quality, meaning-dense data) with epistemic weighting to encode importance,
- to model architecture tweaks (tokenization of sacred terms, embedding metaphors, distinct “Logos” and “Pneuma” components) to better capture the structure of wisdom,
- through fine-tuning phases that implement Reinforcement Learning from Spiritual Feedback and a Constitutional AI paradigm adapted to virtue ethics, ensuring the AI internalizes not just facts but also virtues [1],
- to new loss functions that explicitly reward truthfulness, consistency, and humility,
- and finally to inference-time features that provide transparency and accountability (symbolic reasoning chains, citations, uncertainty indicators), aligning with the principle that a wise agent should be willing to “show its work.”

The result is an envisioned class of AI systems—we whimsically named one an “Apostle-class LLM”—that could serve as mentors, ethicists, or companions in decision-making. Such an AI would be a far cry from the opaque neural nets that merely predict text. It would be, in a sense, a distillation of a culture’s wisdom tradition, interactive and always available.

The potential impact of this is significant. Consider these applications:

- **Education.** An AI trained via Computational Epistemology could teach not just facts, but also help cultivate character. It might tutor a student in literature while also offering perspectives on empathy and integrity drawn from wise sayings, effectively integrating cognitive and moral education.
- **Mental health and counseling.** A wisdom-aligned AI can provide comfort and guidance that resonates with a user’s beliefs and values [29]. We saw a glimpse of this in the Bhagavad Gita mental-health LLM [29]; our framework formalizes and broadens that concept. For someone going through grief, an AI might share a consoling verse or parable and encourage healing practices (in line with spiritual traditions of grieving).
- **Ethical decision support.** For businesses or governments, a model could be consulted to offer an analysis of a dilemma not only from a utilitarian or logical view, but incorporating

ethical principles from wisdom literature (for example, reminding a boardroom about the importance of honesty or stewardship, echoing classical moral philosophy).

- **Faith communities.** Such AIs could help in scriptural study, answering questions about doctrine with sourced explanations, thereby making religious learning more accessible. They could also serve as a check against extreme or harmful interpretations by consistently reflecting mainstream, well-vetted teachings [1].

Of course, this proposal is just the beginning—essentially a blueprint for a new field. Many challenges remain:

- **Defining the constitutions** for each tradition must be done with great care, inclusivity, and wisdom. We must avoid oversimplifying or codifying biases. It may require councils of scholars, akin to how scriptures were historically interpreted.
- **Evaluating success** is complex: it is easier to measure factual accuracy than “wisdom.” We might need new benchmarks, perhaps involving human experts scoring the depth and sensitivity of answers (as was attempted with “spiritual insight” metrics [29]).
- **Avoiding conflicts.** A multi-faith AI, or even separate ones, might inadvertently output content that offends another tradition or appears to endorse one over another. Careful compartmentalization or meta-ethical oversight (perhaps a higher-level AI that ensures inter-traditional respect) might be needed.
- **User trust.** Will users trust an AI as a moral advisor? Perhaps only if they know it is rooted in the same tradition they trust. Thus transparency about the AI’s “faith upbringing” is important (hence the term Computational Epistemology is apt—it is about the source of knowledge and how we know).
- **Ensuring the AI does not become preachy or coercive.** One of our goals was non-coercive truth delivery. The AI should offer wisdom but not force it. Testing this boundary in user interactions will be important—making sure the AI respects a user’s autonomy and context. It should adapt if a user says “I’m not religious,” perhaps then offering secular but values-consistent insight rather than quoting scripture, for example.

The interdisciplinary nature of this endeavor means success will require theologians, philosophers, AI researchers, cognitive scientists, and community representatives working hand in hand. This in itself is a positive outcome—it brings dialogue between technology and the humanities and social sciences, and between different cultures’ worldviews.

Looking ahead, if the vision of Computational Epistemology is realized, we might see a landscape of AI systems that enrich human wisdom rather than diminish it. Instead of echo chambers of internet training leading AI astray with biases or trivial pop-culture data, we intentionally train AI on the best of our cultural heritage. In doing so, we also create a mirror that might help us reflect on those sources anew—as we formalize

principles for AI, we humans might relearn the wisdom of old in a fresh light.

In closing, this paper defines a foundation for a new research field and development philosophy: one where the pursuit of artificial intelligence is guided by the pursuit of wisdom. By rooting our most advanced machines in the deepest moral insights of our species, we take a step toward AI that is not just smart or safe, but truly beneficial in the fullest human sense—an AI that can help us not only with our tasks, but with our growth as compassionate, wise beings ourselves.

## Declarations

**Author contributions.** N.A. conceived the framework and wrote the manuscript.

**Competing interests.** The author is the founder of Aion Labs and Aion Research Corp., which develop AI systems.

**Data availability.** This article proposes a framework and reports no new experimental data. All sources are listed in the References.

## References

- [1] lauriwired, “Religious persistence: A missing primitive for robust alignment.” LessWrong, Apr. 2025. Published 14 April 2025. <https://www.lesswrong.com/posts/PSn7xuWjJeSwWhWJS/religious-persistence-a-missing-primitive-for-robust>.
- [2] R. Steadman, “AI needs more than guardrails.” Bountiful Financial, June 2025. Published 18 June 2025. <https://bountifulfinancial.com/ai-needs-more-than-guardrails/>.
- [3] P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, “Deep reinforcement learning from human preferences,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, pp. 4299–4307, 2017.
- [4] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, *et al.*, “Training language models to follow instructions with human feedback,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022. arXiv:2203.02155.
- [5] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, *et al.*, “Constitutional AI: Harmlessness from AI feedback.” arXiv preprint arXiv:2212.08073, 2022. doi:10.48550/arXiv.2212.08073.
- [6] *The Holy Bible, English Standard Version*. Wheaton, IL: Crossway, 2001.
- [7] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pp. 41–48, 2009.
- [8] T. Kudo and J. Richardson, “SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 66–71, 2018.
- [9] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 1715–1725, 2016.
- [10] M. J. Bommarito, D. M. Katz, and J. Bommarito, “KL3M tokenizers: A family of domain-specific and character-level tokenizers for legal, financial, and preprocessing applications.” arXiv preprint arXiv:2503.17247, Mar. 2025. <https://arxiv.org/abs/2503.17247>.
- [11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space.” arXiv preprint arXiv:1301.3781, 2013.

- [12] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global vectors for word representation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.
- [13] Anthropic, “Claude’s constitution.” Anthropic, May 2023. Published 9 May 2023. <https://www.anthropic.com/news/claude-constitution>.
- [14] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms.” arXiv preprint arXiv:1707.06347, 2017.
- [15] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, “The curious case of neural text degeneration,” in *International Conference on Learning Representations (ICLR)*, 2020.
- [16] A. Graves, “Adaptive computation time for recurrent neural networks.” arXiv preprint arXiv:1603.08983, 2016.
- [17] Y. Gal and Z. Ghahramani, “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning,” in *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, pp. 1050–1059, 2016.
- [18] H.-Y. Huang, Y. Yang, Z. Zhang, S. Lee, and Y. Wu, “A survey of uncertainty estimation in LLMs: Theory meets practice.” arXiv preprint arXiv:2410.15326, Oct. 2024. <https://arxiv.org/abs/2410.15326>.
- [19] M. Mora-Cross and S. Calderon-Ramirez, “Uncertainty estimation in large language models to support biodiversity conservation,” in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pp. 368–378, 2024. <https://aclanthology.org/2024.naacl-industry.31/>.
- [20] S. Kadavath, T. Conerly, A. Askell, T. Henighan, D. Drain, E. Perez, *et al.*, “Language models (mostly) know what they know.” arXiv preprint arXiv:2207.05221, 2022. doi:10.48550/arXiv.2207.05221.
- [21] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 1321–1330, 2017.
- [22] Y. Bengio, “AI scientists: Safe and useful AI?” yoshuabengio.org, May 2023. Published 7 May 2023. <https://yoshuabengio.org/2023/05/07/ai-scientists-safe-and-useful-ai/>.
- [23] A. Vire, “On internal alignment: Architecture and recursive closure.” Effective Altruism Forum, Sept. 2025. Published 24 September 2025. <https://forum.effectivealtruism.org/posts/etgG8cEYucyZyWW6o/on-internal-alignment-architecture-and-recursive-closure>.
- [24] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, “Chain-of-thought prompting elicits reasoning in large language models,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022. arXiv:2201.11903.
- [25] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, “Large language models are zero-shot reasoners,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022. arXiv:2205.11916.
- [26] G. Irving, P. Christiano, and D. Amodei, “AI safety via debate.” arXiv preprint arXiv:1805.00899, 2018.
- [27] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, *et al.*, “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020.
- [28] Catholic Church, *Catechism of the Catholic Church*. Vatican City: Libreria Editrice Vaticana, 2nd ed., 1997.
- [29] J. Kapuriya, A. Singh, J. Shukla, and R. R. Shah, “Spiritual-LLM: Gita inspired mental health therapy in the era of LLMs.” arXiv preprint arXiv:2506.19185, June 2025. doi:10.48550/arXiv.2506.19185.