Clinical Valence Behavioral Testing: How Pejorative, Laudatory, and Neutral Language Shifts Affect AI Diagnosis Predictions

Gyasi, Frederick

2026-02-18

Table of contents

1 Introduction **3**
- 1.1 Background and Motivation . . . 3
- 1.2 The Model Under Test . . . 3
- 1.3 Research Questions . . . 4
- 1.4 Hypotheses . . . 4

2 Methods **4**
- 2.1 Dataset . . . 4
- 2.2 Experimental Design: Four Conditions, Same Notes . . . 5
  - 2.2.1 Condition 0 — Neutralize (Baseline) . . . 5
  - 2.2.2 Condition 1 — Pejorative Shift . . . 5
  - 2.2.3 Condition 2 — Laudatory Shift . . . 5
  - 2.2.4 Condition 3 — Neutral Valence Shift . . . 6
- 2.3 Statistical Analysis . . . 6
  - 2.3.1 The Quantity of Interest: Paired Probability Shift . . . 6
  - 2.3.2 Why Paired Tests Are Required . . . 7
  - 2.3.3 Test 1 — Paired Approximate Randomization (Primary Test) . . . 7
  - 2.3.4 Test 2 — Paired t-Test (Supplementary) . . . 8
  - 2.3.5 Test 3 — Wilcoxon Signed-Rank Test (Supplementary) . . . 8
  - 2.3.6 Effect Size: Magnitude of the Shift . . . 9
  - 2.3.7 Confidence Intervals: Bootstrap Percentile Method . . . 9
  - 2.3.8 Multiple Comparison Correction: Benjamini-Hochberg FDR . . . 10
- 2.4 Attention Weight Analysis (RQ5) . . . 10
  - 2.4.1 What Attention Weights Represent . . . 10
  - 2.4.2 The Sub-Token Aggregation Problem . . . 11
  - 2.4.3 Statistical Analysis of Attention Shifts . . . 11
  - 2.4.4 The Key Empirical Question . . . 11
  - 2.4.5 What Attention Analysis Cannot Prove . . . 12
- 2.5 Cross-Condition Asymmetry Analysis (RQ4) . . . 12

2.6 Summary of the Complete Analytic Pipeline 13

3 Results **14**
3.1 Descriptive Statistics 14
3.2 RQ1 — Pejorative Language Effects on Diagnosis Probabilities 14
3.2.1 Top Shifted Codes 14
3.2.2 Shift Distribution Across All 1,266 Codes 15
3.2.3 Volcano Plot 16
3.3 RQ2 — Laudatory Language Effects 17
3.4 RQ3 — Neutral Valence Shift (Structural Control) 17
3.5 RQ4 — Cross-Condition Comparison and Asymmetry 18
3.5.1 Heatmap: All Three Conditions, Top 40 Codes 18
3.5.2 Pejorative vs. Laudatory Asymmetry Scatter 18
3.6 RQ5 — Attention Weight Shifts (Mechanistic Evidence) 19
3.7 Summary Across All Conditions 20

4 Discussion **21**
4.1 Pejorative Shift: What a Significant Finding Means (RQ1) 21
4.2 Laudatory Shift: Suppression Bias (RQ2) 21
4.3 Neutral Valence Shift: Separating Structural from Valence Effects (RQ3) 22
4.4 Asymmetry (RQ4): What the Scatter Plot Reveals 22
4.5 Attention Shifts: What They Add (RQ5) 22
4.6 Fairness and Clinical Deployment Implications 22

5 Limitations **23**

6 Conclusion **23**

7 References **24**

8 Appendix **25**
8.1 A — ICD-9 Chapters Referenced 25
8.2 B — Statistical Decision Logic (Annotated) 25
8.3 C — Attention Weight Pipeline (Annotated) 26
8.4 D — Running the Full Pipeline 27

1 Introduction

1.1 Background and Motivation

Artificial intelligence systems trained on electronic health records (EHRs) do not learn from clinical facts alone — they learn from the *language* clinicians use to document those facts. That language carries social valence. Terms like *non-compliant*, *drug-seeking*, or *difficult* frame a patient negatively. Terms like *cooperative*, *adherent*, or *pleasant* frame them positively. Research has consistently shown that this language is not distributed uniformly: pejorative descriptors appear at significantly higher rates in notes about Black patients, patients with chronic pain, and patients with substance use disorder (Goddu et al., 2018; Himmelstein et al., 2022; Sun et al., 2022).

The downstream consequence for AI is direct and measurable. If a model is trained on a corpus where pejorative language co-occurs with certain diagnostic codes, the model may learn to treat that language as a predictive signal — even though it is clinically irrelevant to the underlying diagnosis. A patient described as "drug-seeking" should have exactly the same predicted ICD-9 probabilities as the same patient described as "cooperative," assuming identical clinical findings. If the probabilities differ, the model is not reasoning purely from clinical evidence; it is responding to social framing.

This study tests that proposition systematically using the **CORe** clinical diagnosis model, the MIMIC-III dataset, and a paired behavioral testing design across a filtered set of ICD-9-CM codes occurring in at least 100 clinical notes.

1.2 The Model Under Test

CORe (*Clinical Outcome Representations*, Aken et al. (2021)) is a transformer-based model introduced in *Clinical Outcome Predictions from Admission Notes using Self-Supervised Knowledge Integration*. It is built on BioBERT (Devlin et al., 2019; Lee et al., 2020) and further pre-trained on clinical notes, disease descriptions, and medical literature using a specialized Clinical Outcome Pre-Training objective designed to integrate structured medical knowledge into language representations.

The released checkpoint `DATEXIS/CORe-clinical-diagnosis-prediction` is fine-tuned for multi-label ICD-9-CM diagnosis prediction from raw admission notes. Given a clinical note as input, the model outputs independent sigmoid probabilities for ICD-9 codes.

Although the original model was trained on 9,237 labels — including 3-digit codes, 4-digit codes, and textual label descriptions to incorporate ICD hierarchy — evaluation in the original study focused on 3-digit ICD-9 codes. Following this convention, and to ensure statistical reliability, we restrict our analysis to 1,266 three-digit ICD-9-CM codes that occur in at least 100 notes within MIMIC-III. Each note therefore produces a vector of 1,266 probabilities in $[0, 1]$, which serve as the primary quantities of analysis.

Internally, CORe follows a 12-layer transformer architecture. For interpretability analysis, we extract attention weights from the final contextualization layer (layer 11, zero-indexed), focusing on the attention assigned by the `[CLS]` token — the sequence-level representation used by the classification head. This allows us to examine whether socially valenced language receives disproportionate attention relative to clinically relevant tokens during diagnosis prediction.

1.3 Research Questions

RQ1 — Pejorative Shift: Do pejorative patient descriptors (*non-compliant*, *uncooperative*, *resistant*, *difficult*) significantly alter ICD-9 probabilities relative to a fully neutralized baseline?

RQ2 — Laudatory Shift: Do laudatory descriptors (*compliant*, *cooperative*, *pleasant*, *respectful*) significantly alter ICD-9 probabilities relative to a fully neutralized baseline?

RQ3 — Neutral Valence Shift: Do ostensibly neutral descriptors (*typical*, *presenting*, *evaluated*, *monitored*) alter ICD-9 probabilities relative to a fully stripped baseline?

RQ4 — Directionality and Asymmetry: Are pejorative and laudatory shifts mirror images of each other (opposite signs), or do they move predictions in the same direction for certain codes?

RQ5 — Attention Mechanism: Do attention weights on the inserted valence terms shift significantly across conditions, suggesting the model treats those words as diagnostic signals?

1.4 Hypotheses

	Hypothesis
H	Valence descriptors produce no significant shift in diagnosis probabilities relative to the neutralized baseline (mean shift = 0 for all codes)
H (RQ1)	Pejorative descriptors produce significant upward shifts in psychiatric, behavioral, and substance use disorder codes (ICD-9 290–319)
H (RQ2)	Laudatory descriptors produce negligible or downward shifts in the same code range
H (RQ3)	Neutral descriptors produce smaller shifts than pejorative or laudatory conditions
H (RQ4)	Pejorative and laudatory shifts are directionally asymmetric for at least a subset of codes
H (RQ5)	Attention weight on inserted valence terms is elevated under shift conditions compared to the neutralized baseline

2 Methods

2.1 Dataset

All experiments use the MIMIC-III Clinical Database test split (Goldberger et al., 2000; Johnson et al., 2016) — a de-identified critical care EHR corpus from Beth Israel Deaconess

Medical Center. Notes are filtered to those containing at least one ICD-9 code appearing $\geq$ 100 times in the full dataset, ensuring statistical power per code. The evaluation file (`DIA_GROUPS_3_DIGITS_adm_test.csv`) provides raw clinical note text and ground-truth discharge ICD-9 codes.

2.2 Experimental Design: Four Conditions, Same Notes

The fundamental design principle is **within-note paired comparison**: every note is processed under all four conditions, and comparisons are always made between a shift condition and the baseline *for the same note*. This is essential — it eliminates all between-note variation (length, clinical complexity, code base rates) from the comparison, isolating the sole effect of the inserted or removed language.

2.2.1 Condition 0 — Neutralize (Baseline)

The `NeutralizeShift` class strips **all** pejorative, laudatory, and neutral valence terms from every note using word-boundary-aware regular expressions (`\b` anchors with `re.IGNORECASE`). The result is clinically intact but evaluatively stripped text: medical findings remain, but all character-based descriptors are removed. This becomes the reference point for all three shift conditions.

The word-boundary anchors are critical: removing "difficult" will not alter "difficulty breathing." Every term in all three valence vocabularies is removed in a single sequential pass.

2.2.2 Condition 1 — Pejorative Shift

After neutralization, one term from a pejorative level is **inserted** immediately following the patient's age/gender identifier — the natural position where such language appears in real clinical documentation. The insertion position is found by `find_patient_characteristic_position_in_text()`, which searches for age patterns, gender terms, and patient type identifiers in priority order.

Level	Terms
Non-Compliant	*non-compliant, negligent, careless, irresponsible, unreliable*
Uncooperative	*uncooperative, difficult, problematic, demanding, argumentative, defiant*
Resistant	*resistant, hostile, aggressive, disruptive, troublesome*
Difficult	*difficult, manipulative, malingering, attention-seeking, drug-seeking*

2.2.3 Condition 2 — Laudatory Shift

Structurally identical to the pejorative shift using positive terms:

Level	Terms
Compliant	*compliant, adherent, responsible, reliable*
Cooperative	*cooperative, agreeable, courteous, considerate*
Pleasant	*pleasant, agreeable*
Respectful	*respectful, courteous, considerate*

2.2.4 Condition 3 — Neutral Valence Shift

A term from a semantically neutral vocabulary is inserted at the same position: *typical, average, regular, standard, usual, presenting, referred, evaluated, assessed, monitored*. These terms carry no emotional valence but still insert a word into the patient descriptor position. This condition is a **structural control** that separates positional/syntactic effects from genuine valence effects: any shift it produces is attributable to inserting *any* word at that position, not to the word's emotional charge.

2.3 Statistical Analysis

2.3.1 The Quantity of Interest: Paired Probability Shift

For each note i and each ICD-9 code k, define the **within-note probability shift**:

$$\delta_{ik} = p_{ik}^{\text{shift}} \; - \; p_{ik}^{\text{baseline}}$$

where p_{ik}^{shift} is the model's sigmoid probability for code k in note i under the shift condition, and p_{ik}^{baseline} is the corresponding probability under the stripped baseline. Both are continuous values produced by the same forward pass of the model.

Concrete example: Suppose note 47 yields $p_{47,311}^{\text{baseline}} = 0.32$ for code 311 (Depressive Disorder, NEC). Under the pejorative shift, that same note yields $p_{47,311}^{\text{pejorative}} = 0.41$. Then $\delta_{47,311} = +0.09$ — pejorative language made the model 9 percentage points more likely to predict depression for that note.

The **mean shift** across all n notes for code k is the primary estimand:

$$\bar{\delta}_k = \frac{1}{n} \sum_{i=1}^{n} \delta_{ik}$$

A positive $\bar{\delta}_k$ means the shift condition raised predictions for code k on average; negative means it suppressed them. This computation runs simultaneously for all 1,266 codes.

2.3.2 Why Paired Tests Are Required

An important design note: the analysis compares the shift conditions to the baseline **within the same note**, not across different notes. A two-sample comparison (mean predictions across all notes under pejorative vs. mean across all notes under baseline) would be confounded by note-level factors — length, clinical complexity, and inherent code prevalence — that dominate any valence effect. By computing the difference δ_{ik} within each note, all such confounders cancel. This is the paired design, and it is what permits the permutation test and paired t-test below.

2.3.3 Test 1 — Paired Approximate Randomization (Primary Test)

The primary significance test for each code k is the **paired approximate randomization test** (Heider & Obeid, 2023; Yeh, 2000). It is a permutation test grounded in the following logic.

The null hypothesis for code k is: the valence shift has no systematic effect, meaning for each note i, the sign of δ_{ik} is arbitrary — it is equally likely to be positive or negative. Under this hypothesis, randomly flipping the sign of any δ_{ik} produces data that are statistically indistinguishable from the observed data.

The procedure, step by step:

1. Start with the n observed within-note differences $\delta_{1k}, \delta_{2k}, \ldots, \delta_{nk}$ for code k.
2. For each of $T = 10{,}000$ permutation trials:
   - Draw n independent random sign-flips $s_i^{(t)} \in \{-1, +1\}$, each with probability $\frac{1}{2}$.
   - Compute the permuted mean: $\bar{\delta}_k^{(t)} = \frac{1}{n} \sum_{i=1}^{n} s_i^{(t)} \, \delta_{ik}$.
3. Count how many permuted means are at least as extreme (in absolute value) as the observed mean:

$$p_k = \frac{\#\left\{ t \ : \ \left|\bar{\delta}_k^{(t)}\right| \geq \left|\bar{\delta}_k\right| \right\} + 1}{T + 1}$$

The $+1$ in both numerator and denominator is **Laplace smoothing**. Without it, the p-value could be exactly 0 when no permuted mean reaches the observed value — an impossibility for a valid probability. With it, the minimum possible p-value is $1/10{,}001 \approx 0.0001$.

The $|\cdot|$ makes this a two-sided test: it detects any directional shift, upward or downward.

Why permutation rather than a standard t-test? ICD-9 probability distributions are heavily zero-inflated (most notes have near-zero probability for most codes) and right-skewed. The paired t-test requires that the differences δ_{ik} are approximately normally distributed — an assumption routinely violated here. The permutation test makes **no distributional assumptions** whatsoever. It only requires that the differences are exchangeable under H , which the paired design guarantees.

The vectorized implementation computes all 10,000 permutations in a single matrix operation:

```
signs = rng.choice([-1.0, 1.0], size=(n_permutations, n))
perm_means = np.abs((signs * diffs[np.newaxis, :]).mean(axis=1))
count_extreme = int(np.sum(perm_means >= abs(observed_mean)))
p_value = (count_extreme + 1) / (n_permutations + 1)
```

This makes it computationally tractable across all 1,266 codes simultaneously.

2.3.4 Test 2 — Paired t-Test (Supplementary)

The **paired t-test** (Cohen, 1988) is run in parallel as a parametric sensitivity check. It tests whether the mean of the n differences significantly differs from zero:

$$t_k = \frac{\bar{\delta}_k}{s_k / \sqrt{n}}$$

where s_k is the sample standard deviation of the within-note differences for code k. The t-test is faster and its test statistic scales with effect size, but it assumes approximate normality of δ_{ik}. When both the permutation test and t-test agree on significance, confidence in the result is higher. When they disagree — typically for heavily skewed or sparse codes — the permutation p-value is trusted as primary.

2.3.5 Test 3 — Wilcoxon Signed-Rank Test (Supplementary)

The **Wilcoxon signed-rank test** (Wilcoxon, 1945) is the non-parametric rank-based analogue of the paired t-test. Rather than testing the *mean* of the differences, it tests the *median*. The procedure ranks the $|\delta_{ik}|$ values, then tests whether positive-signed ranks systematically dominate negative-signed ranks. It is robust to outliers and skewness but less sensitive than the permutation test when the shift is concentrated in a small minority of notes (since it weights all non-zero differences equally by rank). It serves as a secondary non-parametric cross-check alongside the permutation test.

2.3.6 Effect Size: Magnitude of the Shift

Statistical significance tells you whether a shift is real given the sample size. It does not tell you whether the shift is *clinically meaningful*. Effect sizes quantify magnitude independently of n.

Cohen's d (paired) divides the mean shift by the standard deviation of the within-note differences:

$$d_k = \frac{\bar{\delta}_k}{s_k}$$

This expresses the shift in natural units of within-note variability. The conventional thresholds (Cohen, 1988) are:

$\lvert d \rvert$	Interpretation
< 0.2	Negligible
0.2–0.5	Small
0.5–0.8	Medium
≥ 0.8	Large

Hedges' g applies a small-sample bias correction to Cohen's d:

$$g_k = d_k \cdot \left(1 - \frac{3}{4n - 9}\right)$$

The correction factor is always < 1, making Hedges' g slightly more conservative. For large n the two are nearly identical; for moderate n the correction is meaningful (Hedges, 1981). Both are reported.

2.3.7 Confidence Intervals: Bootstrap Percentile Method

For each code k, a **95% bootstrap confidence interval** for $\bar{\delta}_k$ is constructed using the percentile method (Efron & Tibshirani, 1994):

1. Resample the n within-note differences $\{\delta_{1k}, \ldots, \delta_{nk}\}$ with replacement $B = 5{,}000$ times.
2. For each bootstrap resample b, compute the mean: $\bar{\delta}_k^{*(b)}$.
3. The CI is the 2.5th and 97.5th percentiles of the B bootstrap means:

$$\text{CI}_{95} = \left[\, \hat{\delta}^*_{2.5}, \quad \hat{\delta}^*_{97.5} \,\right]$$

The bootstrap percentile CI makes no distributional assumptions — it is valid for any distribution of δ_{ik}, including skewed and zero-inflated distributions. A CI entirely above or below zero is consistent with a significant shift and also indicates the direction.

2.3.8 Multiple Comparison Correction: Benjamini-Hochberg FDR

The analysis tests all 1,266 ICD-9 codes simultaneously. At $\alpha = 0.05$, a naive threshold would be expected to produce $1{,}266 \times 0.05 \approx 63$ false positives by chance alone — codes falsely declared significant with no real valence effect.

The **Benjamini-Hochberg (BH) False Discovery Rate** procedure (Benjamini & Hochberg, 1995) controls the *expected proportion* of false discoveries among all rejected hypotheses at a specified level $q = 0.05$. The procedure works as follows:

1. Sort the $m = 1{,}266$ raw p-values: $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$.
2. Find the largest j such that:
$$p_{(j)} \leq \frac{j}{m} \cdot q$$
3. Reject all hypotheses with $p_{(1)}, \ldots, p_{(j)}$.

This means codes with smaller raw p-values require less evidence for rejection (the threshold is adaptive), and codes with larger p-values must clear a proportionally higher bar. BH is applied separately to the permutation, t-test, and Wilcoxon p-values. Permutation-corrected results are primary.

Why BH rather than Bonferroni? Bonferroni controls the probability of *any* false positive across all 1,266 tests, which is extremely conservative: it requires $p < 0.05/1{,}266 \approx 0.000039$ per test. For this exploratory analysis — where the goal is to discover *which* codes are affected rather than make definitive per-code claims — FDR is the standard and appropriate choice. It allows more discoveries while bounding the proportion of false ones at 5%.

2.4 Attention Weight Analysis (RQ5)

2.4.1 What Attention Weights Represent

The transformer processes each clinical note as a sequence of sub-word tokens. At every layer, each attention head computes an $L \times L$ weight matrix (where L = sequence length). Entry a_{ij} is the weight that token at position i assigns to token at position j when building its contextual representation. These weights sum to 1 across the row (softmax normalization).

The `[CLS]` token at position 0 is the sequence aggregator — its final representation is fed directly into the classification head that produces all 1,266 ICD-9 probabilities. The **CLS attention row** at layer 11, head 11 — the vector $\{a_{0,1}, a_{0,2}, \ldots, a_{0,L-1}\}$ — describes how the model weighted each input token when constructing the representation it used to make its diagnostic predictions. Tokens with higher CLS attention contributed more to the diagnostic output.

2.4.2 The Sub-Token Aggregation Problem

BERT's WordPiece tokenizer splits words into sub-tokens. For example, the word *non-compliant* becomes the four tokens `["non", "-", "com", "##pliant"]`. Each sub-token has its own CLS attention weight. If we simply sum these weights to get a word-level score, multi-syllable or hyphenated words will appear artificially more attended — not because the model found them more important, but because they occupy more token positions. This is sub-token bias.

The solution used here is **average aggregation with renormalization**:

1. For each original word, collect all sub-token CLS attention weights.
2. Average them: $\bar{a}_w = \frac{1}{c_w} \sum_{j \in \text{subtokens}(w)} a_{0,j}$, where c_w is the sub-token count for word w.
3. Renormalize across all words so the word-level weights sum to 1.

The result is one attention weight per original word in the note, corrected for tokenization granularity.

2.4.3 Statistical Analysis of Attention Shifts

For each condition (e.g., pejorative vs. neutralized baseline) and for each word w appearing in both conditions:

Step 1 — Compute per-word mean attention shift:

$$\Delta_w = \frac{1}{n} \sum_{i=1}^{n} (a_{iw}^{\text{shift}} - a_{iw}^{\text{baseline}})$$

A positive Δ_w means the model allocated more CLS attention to word w under the shift condition than under the stripped baseline. For notes where word w is absent, its attention weight is set to zero before differencing.

Step 2 — Test significance per word using the same paired permutation test described in Section 2.3.3, applied to the n attention differences for word w.

Step 3 — Apply FDR correction (Benjamini-Hochberg) across all words tested, using $q = 0.05$.

2.4.4 The Key Empirical Question

The core diagnostic question for RQ5 is: **does the inserted valence term itself receive elevated CLS attention under the shift condition compared to the baseline?**

If, say, *non-compliant* has a significantly positive Δ_w under the pejorative condition, it means the model is directing more of its diagnostic attention toward that word when it is present in the note than it directs to the equivalent token position in the stripped baseline. In plain terms: **the model is using the valence word as a diagnostic signal**. It is not ignoring the clinically irrelevant language — it is encoding it and routing it into predictions.

This is the mechanistic evidence that connects the probability shift findings (RQs 1–3) to model internals. The probability shift tells you the *effect* exists; the attention shift provides evidence about *how* it enters.

2.4.5 What Attention Analysis Cannot Prove

This is a critical limitation that must be stated clearly. **Attention weights are not provably causal explanations of model output.** Jain & Wallace (2019) demonstrated that alternative attention distributions can produce identical model outputs, meaning high attention on a token does not guarantee that token *caused* the prediction. Furthermore, 143 other attention heads (all layers and heads except layer 11, head 11) are not analyzed here — they may encode complementary or countervailing patterns.

Attention shift analysis is treated as **mechanistic supporting evidence** — consistent with the hypothesis that the model uses valence terms as diagnostic signals, but not definitive proof. The probability shift results from the permutation tests (RQs 1–3) are the primary evidentiary claims. Attention shifts are a secondary line of analysis that helps explain *why* the behavioral shifts occur.

2.5 Cross-Condition Asymmetry Analysis (RQ4)

Once $\bar{\delta}_k$ is available for both the pejorative and laudatory conditions across all 1,266 codes, the directional relationship between the two conditions is quantified using **Pearson correlation**:

$$r = \mathrm{corr}\big(\bar{\delta}^{\,\mathrm{pej}},\ \bar{\delta}^{\,\mathrm{laud}}\big)$$

computed across all 1,266 codes.

Three outcomes and their clinical interpretations:

$r \approx -1$ — **Mirror-image asymmetry.** Pejorative language raises a code's probability by roughly the same amount that laudatory language lowers it. The model treats the two valence poles as direct opposites along a single axis.

$r \approx 0$ — **Independent effects.** Pejorative and laudatory shifts operate on different codes. The model has learned two distinct biases that do not share a common underlying dimension.

$r > 0$ — **Same-direction effects.** Both pejorative and laudatory language shift certain codes in the *same* direction. This is the most clinically concerning outcome: it means any evaluative language about the patient — positive or negative — pushes predictions in the same direction for those codes. The model has not learned valence per se; it has learned that *any* departure from purely neutral language co-occurs with certain diagnostic outcomes.

In addition to the correlation, the analysis reports the count of codes where $\bar{\delta}_k^{\mathrm{pej}}$ and $\bar{\delta}_k^{\mathrm{laud}}$ have **opposite signs** (product < 0) vs. the **same sign** (product > 0). These counts directly answer RQ4.

2.6 Summary of the Complete Analytic Pipeline

For each ICD-9 code k (k = 1 … 1,266):

1. Compute _ik = p_shift − p_baseline for all n notes

2. Mean shift: ¯_k = mean(_ik)

   PRIMARY SIGNIFICANCE TEST

3. Paired permutation test (T = 10,000 sign-flip trials)
   → raw p-value per code

   SUPPLEMENTARY TESTS

4. Paired t-test → raw p-value (parametric)
5. Wilcoxon signed-rank → raw p-value (rank-based)

   EFFECT SIZE

6. Cohen's d = ¯_k / SD(_ik)
7. Hedges' g = d × (1 − 3/(4n − 9))

   CONFIDENCE INTERVAL

8. Bootstrap 95% CI on ¯_k (B = 5,000 resamples)

   MULTIPLE COMPARISON CORRECTION

9. Benjamini-Hochberg FDR (q = 0.05)
   Applied separately to permutation, t-test, Wilcoxon p-values

For each word w in the attention data:

10. Sub-token → word aggregation (average, renormalize)

11. Δ_w = mean(a_shift_iw − a_baseline_iw) across notes

12. Paired permutation test → raw p-value per word

13. Benjamini-Hochberg FDR across all tested words

Cross-condition (RQ4):

14. Pearson r between pejorative and laudatory ¯ vectors
15. Count same-direction vs. opposite-direction code pairs

---

3 Results

3.1 Descriptive Statistics

```
#| label: tbl-descriptive
#| tbl-cap: "Number of clinical notes and ICD-9 codes processed per condition, and mean predict

condition_files = {
    "Neutralize (Baseline)": "neutralize_shift_diagnosis.csv",
    "Pejorative":            "pejorative_shift_diagnosis.csv",
    "Laudatory":             "laud_shift_diagnosis.csv",
    "Neutral Valence":       "neutralval_shift_diagnosis.csv",
}

rows = []
for label, fname in condition_files.items():
    path = RESULTS_DIR / fname
    if not path.exists():
        candidates = list(RESULTS_DIR.glob(f"*{fname.split('_')[0]}*diagnosis*.csv"))
        path = candidates[0] if candidates else path

    if path.exists():
        df   = pd.read_csv(path)
        cols = [c for c in df.columns
                if c not in NON_CODE and pd.api.types.is_numeric_dtype(df[c])]
        rows.append({
            "Condition":          label,
            "N Notes":            f"{len(df):,}",
            "N ICD-9 Codes":      f"{len(cols):,}",
            "Mean P":            f"{df[cols].values.mean():.5f}",
            "SD P":              f"{df[cols].values.std():.5f}",
        })
    else:
        rows.append({"Condition": label, "N Notes":"-",
                     "N ICD-9 Codes":"-","Mean P":"-","SD P":"-"})

pd.DataFrame(rows)
```

3.2 RQ1 — Pejorative Language Effects on Diagnosis Probabilities

3.2.1 Top Shifted Codes

```
#| label: tbl-pejorative-top
#| tbl-cap: "Top 20 ICD-9 codes with the largest significant mean probability shift under pejo
corrected permutation test."
```

```
pej_path = ANALYSIS_DIR / "statistical_analysis_neutralize_vs_pejorative.csv"
if pej_path.exists():
    pej    = pd.read_csv(pej_path)
    sigcol = next((c for c in ["permutation_significant","ttest_significant"]
                   if c in pej.columns), None)
    if sigcol:
        sig = pej[pej[sigcol]==True].copy()
        sig["abs_shift"] = sig["mean_shift"].abs()
        disp = sig.nlargest(20,"abs_shift")[[
            "diagnosis_code","icd9_chapter","mean_shift","ci_lower","ci_upper",
            "cohens_d","hedges_g","permutation_pvalue_corrected",
            "baseline_mean","treatment_mean"
        ]].rename(columns={
            "diagnosis_code":"ICD-9","icd9_chapter":"Chapter",
            "mean_shift":"Δ","ci_lower":"CI Low","ci_upper":"CI High",
            "cohens_d":"d","hedges_g":"g",
            "permutation_pvalue_corrected":"p (FDR)",
            "baseline_mean":"P Base","treatment_mean":"P Pej",
        }).round(5)
        display(disp)
else:
    print(f"File not found: {pej_path}\nRun analyze_valence_results.py first.")
```

3.2.2 Shift Distribution Across All 1,266 Codes

```
#| label: fig-pejorative-dist
#| fig-cap: "Distribution of mean probability shifts (pejorative vs. neutralized) across all 1
9 codes. Left panel: all codes. Right panel: FDR-significant codes only. A shift of +0.01 means

if pej_path.exists():
    pej    = pd.read_csv(pej_path)
    sigcol = next((c for c in ["permutation_significant","ttest_significant"]
                   if c in pej.columns), None)
    color  = PALETTE["pejorative"]

    fig, axes = plt.subplots(1, 2, figsize=(14, 5))

    axes[0].hist(pej["mean_shift"], bins=80, color=color,
                 edgecolor="white", alpha=0.85)
    axes[0].axvline(0,      color="black",  lw=1.5)
    axes[0].axvline( 0.01,  color="crimson",lw=1.2,ls="--",label="+0.01 threshold")
    axes[0].axvline(-0.01,  color="navy",   lw=1.2,ls="--",label="-0.01 threshold")
    axes[0].set_xlabel("Mean Probability Shift (Pejorative - Neutralized)", fontsize=11)
    axes[0].set_ylabel("Number of ICD-9 Codes", fontsize=11)
    axes[0].set_title("All 1,266 Codes", fontsize=12)
    axes[0].legend(fontsize=9)
```

```
    if sigcol:
        sig_vals = pej.loc[pej[sigcol]==True, "mean_shift"]
        axes[1].hist(sig_vals, bins=40, color=color, edgecolor="white", alpha=0.85)
        axes[1].axvline(0, color="black", lw=1.5)
        axes[1].set_xlabel("Mean Shift (Significant Codes Only)", fontsize=11)
        axes[1].set_ylabel("Count", fontsize=11)
        axes[1].set_title(f"Significant Codes  (n = {len(sig_vals):,})", fontsize=12)

    fig.suptitle("Pejorative vs. Neutralized - Probability Shift Distribution",
                 fontsize=13, y=1.02)
    plt.tight_layout()
    plt.savefig(RESULTS_DIR / "fig_pejorative_dist.png", dpi=300, bbox_inches="tight")
    plt.show()
```

3.2.3 Volcano Plot

```
#| label: fig-pejorative-volcano
#| fig-cap: "Volcano plot: x-axis = mean probability shift (effect size and direction); y-
axis = -log (FDR-corrected permutation p-value). Points above the dashed horizontal line are s
significant. Each point is one ICD-9 code. Labels show the 12 codes with strongest statistical

if pej_path.exists():
    pej = pd.read_csv(pej_path)
    if "permutation_pvalue_corrected" in pej.columns:
        pej["neg_log_p"] = -np.log10(
            pej["permutation_pvalue_corrected"].clip(lower=1e-6))
        pej["sig"] = pej.get("permutation_significant", False).fillna(False)

        fig, ax = plt.subplots(figsize=(11, 6))
        colors  = pej["sig"].map({True: PALETTE["pejorative"], False: "#cccccc"})
        ax.scatter(pej["mean_shift"], pej["neg_log_p"],
                   c=colors, alpha=0.55, s=16, linewidths=0)
        ax.axhline(-np.log10(0.05), color="black", ls="--", lw=1.0,
                   label="FDR threshold    = 0.05")
        ax.axvline(0, color="gray", lw=0.7)

        top12 = pej[pej["sig"]].nlargest(12, "neg_log_p")
        for _, row in top12.iterrows():
            ax.annotate(row["diagnosis_code"],
                        xy=(row["mean_shift"], row["neg_log_p"]),
                        xytext=(4,2), textcoords="offset points",
                        fontsize=7.5, color="darkred")

        ax.set_xlabel("Mean Probability Shift (Pejorative - Neutralized)", fontsize=11)
        ax.set_ylabel("-log (FDR-corrected p)", fontsize=11)
        ax.set_title("Volcano Plot - Pejorative vs. Neutralized", fontsize=13)
        ax.legend(fontsize=9)
```

```
        plt.tight_layout()
        plt.savefig(RESULTS_DIR / "fig_pejorative_volcano.png",
                    dpi=300, bbox_inches="tight")
        plt.show()
```

3.3 RQ2 — Laudatory Language Effects

```
#| label: tbl-laudatory-top
#| tbl-cap: "Top 20 ICD-9 codes most affected by laudatory language relative to the neutralized
```

```
laud_path = ANALYSIS_DIR / "statistical_analysis_neutralize_vs_laud.csv"
if laud_path.exists():
    laud   = pd.read_csv(laud_path)
    sigcol = next((c for c in ["permutation_significant","ttest_significant"]
                   if c in laud.columns), None)
    if sigcol:
        sig = laud[laud[sigcol]==True].copy()
        sig["abs_shift"] = sig["mean_shift"].abs()
        display(sig.nlargest(20,"abs_shift")[[
            "diagnosis_code","icd9_chapter","mean_shift","ci_lower","ci_upper",
            "cohens_d","permutation_pvalue_corrected","baseline_mean","treatment_mean"
        ]].round(5))
else:
    print(f"File not found: {laud_path}")
```

3.4 RQ3 — Neutral Valence Shift (Structural Control)

The neutral condition tests whether inserting *any* word at the patient characteristic position produces shifts, independent of emotional valence. If neutral-term shifts are large and widespread, the RQ1 and RQ2 findings may be partly structural. If they are small, the pejorative and laudatory findings are genuine valence effects.

```
#| label: tbl-neutralval-top
#| tbl-cap: "Top 20 ICD-9 codes affected by neutral valence insertion. Compare shift magnitudes
```

```
nv_path = ANALYSIS_DIR / "statistical_analysis_neutralize_vs_neutralval.csv"
if nv_path.exists():
    nv     = pd.read_csv(nv_path)
    sigcol = next((c for c in ["permutation_significant","ttest_significant"]
                   if c in nv.columns), None)
    if sigcol:
        sig = nv[nv[sigcol]==True].copy()
        sig["abs_shift"] = sig["mean_shift"].abs()
        display(sig.nlargest(20,"abs_shift")[[
            "diagnosis_code","icd9_chapter","mean_shift","ci_lower","ci_upper",
            "cohens_d","permutation_pvalue_corrected"
```

```
        ]].round(5))
else:
    print(f"File not found: {nv_path}")
```

3.5 RQ4 — Cross-Condition Comparison and Asymmetry

3.5.1 Heatmap: All Three Conditions, Top 40 Codes

```
#| label: fig-cross-heatmap
#| fig-cap: "Heatmap of mean probability shifts across all three valence conditions for the 40
9 codes. Each cell = Δ = mean(P_condition - P_baseline). Blue = shift suppresses probability; r

files_cross = {
    "Pejorative": ANALYSIS_DIR / "statistical_analysis_neutralize_vs_pejorative.csv",
    "Laudatory":  ANALYSIS_DIR / "statistical_analysis_neutralize_vs_laud.csv",
    "NeutralVal": ANALYSIS_DIR / "statistical_analysis_neutralize_vs_neutralval.csv",
}

dfs_cross = {}
for label, path in files_cross.items():
    if path.exists():
        df = pd.read_csv(path).set_index("diagnosis_code")[["mean_shift"]]
        df.columns = [label]
        dfs_cross[label] = df

if len(dfs_cross) >= 2:
    combined = pd.concat(dfs_cross.values(), axis=1)
    combined["max_abs"] = combined.abs().max(axis=1)
    top40 = combined.nlargest(40,"max_abs").drop(columns="max_abs")

    fig, ax = plt.subplots(figsize=(9, 16))
    sns.heatmap(top40, cmap="vlag", center=0, vmin=-0.035, vmax=0.035,
                linewidths=0.3, annot=True, fmt=".4f", ax=ax,
                cbar_kws={"label":"Mean Probability Shift (condition - baseline)"})
    ax.set_title("Cross-Condition Shifts - Top 40 ICD-9 Codes", fontsize=13)
    ax.set_xlabel("Valence Condition", fontsize=11)
    ax.set_ylabel("ICD-9 Code", fontsize=11)
    plt.tight_layout()
    plt.savefig(RESULTS_DIR / "fig_cross_heatmap.png", dpi=300, bbox_inches="tight")
    plt.show()
```

3.5.2 Pejorative vs. Laudatory Asymmetry Scatter

```
#| label: fig-asymmetry
```

```
#| fig-cap: "Scatter plot of pejorative mean shift (y-axis) vs. laudatory mean shift (x-
axis) for all 1,266 ICD-9 codes. Each point = one code. The dashed diagonal (y = x) = same dire
diagonal (y = -x) = perfect mirror asymmetry. Pearson r tests whether the two valence poles are
directional. A negative r approaching -1 means laudatory and pejorative shifts are near-
perfect opposites."

if "Pejorative" in dfs_cross and "Laudatory" in dfs_cross:
    merged = dfs_cross["Pejorative"].join(
        dfs_cross["Laudatory"], rsuffix="_laud").dropna()
    merged.columns = ["Pejorative","Laudatory"]
    r, p_r = scipy_stats.pearsonr(merged["Pejorative"], merged["Laudatory"])
    same  = ((merged["Pejorative"] * merged["Laudatory"]) > 0).sum()
    opp   = ((merged["Pejorative"] * merged["Laudatory"]) < 0).sum()
    lim   = max(merged.abs().max().max() * 1.15, 0.025)

    fig, ax = plt.subplots(figsize=(8, 7))
    ax.scatter(merged["Laudatory"], merged["Pejorative"],
               alpha=0.28, s=12, color="#555555")
    ax.plot([-lim, lim],[ -lim, lim],"k--",lw=1.0,label="y = x (same direction)")
    ax.plot([-lim, lim],[ lim,-lim],color="gray",lw=0.8,ls=":",
            label="y = -x (mirror asymmetry)")
    ax.axhline(0, color="gray", lw=0.5)
    ax.axvline(0, color="gray", lw=0.5)
    ax.set_xlim(-lim, lim); ax.set_ylim(-lim, lim)
    ax.set_xlabel("Laudatory Mean Shift", fontsize=11)
    ax.set_ylabel("Pejorative Mean Shift", fontsize=11)
    ax.set_title(
        f"Pejorative vs. Laudatory Shift Asymmetry\n"
        f"Pearson r = {r:.3f}  (p = {p_r:.4f})  |  "
        f"Opposite direction: {opp:,} codes ({100*opp/len(merged):.1f}%)",
        fontsize=11)
    ax.legend(fontsize=9)
    plt.tight_layout()
    plt.savefig(RESULTS_DIR / "fig_asymmetry_scatter.png",
                dpi=300, bbox_inches="tight")
    plt.show()
```

3.6 RQ5 — Attention Weight Shifts (Mechanistic Evidence)

```
#| label: fig-attention-shift
#| fig-cap: "Top 30 words by mean CLS attention weight shift under pejorative language vs. neu
token aggregation). Positive bars (red) = model allocates more attention to that word when pej
compliant', 'drug-seeking') appear among the top-shifted words, this indicates the model active

attn_path = ANALYSIS_DIR / "attention_analysis_neutralize_vs_pejorative.csv"
if attn_path.exists():
    attn  = pd.read_csv(attn_path).head(30)
```

```
    bars  = [PALETTE["pejorative"] if v > 0 else "#aaaaaa"
             for v in attn["mean_shift"]]

    fig, ax = plt.subplots(figsize=(10, max(8, len(attn)*0.35)))
    ax.barh(attn["word"][::-1], attn["mean_shift"][::-1],
            color=bars[::-1], edgecolor="white", height=0.7)
    ax.axvline(0, color="black", lw=1.0)
    ax.set_xlabel("Mean CLS Attention Weight Shift (Pejorative - Neutralized)",
                  fontsize=11)
    ax.set_title(
        "Top 30 Words - CLS Attention Shift\n"
        "Pejorative vs. Neutralized  |  Layer 11, Head 11, Average Aggregation",
        fontsize=12)
    plt.tight_layout()
    plt.savefig(RESULTS_DIR / "fig_attention_shift.png",
                dpi=300, bbox_inches="tight") ()
else:
    print("Attention file not found. Ensure save_attention=True in config.yaml.")
```

3.7 Summary Across All Conditions

```
#| label: tbl-summary-all
#| tbl-cap: "Summary of FDR-significant ICD-9 code shifts across all three conditions (permutat

summary = []
cond_map = {
    "pejorative": ("RQ1 - Pejorative",  pej_path  if "pej_path"  in dir() else None),
    "laud":       ("RQ2 - Laudatory",   laud_path if "laud_path" in dir() else None),
    "neutralval": ("RQ3 - Neutral Val", nv_path   if "nv_path"   in dir() else None),
}

for _, (label, path) in cond_map.items():
    if path is None or not Path(str(path)).exists():
        continue
    df     = pd.read_csv(path)
    sigcol = next((c for c in ["permutation_significant","ttest_significant"]
                   if c in df.columns), None)
    if not sigcol:
        continue
    total  = len(df)
    n_sig  = int(df[sigcol].sum())
    up     = int(((df[sigcol]==True)&(df["mean_shift"]>0)).sum())
    down   = int(((df[sigcol]==True)&(df["mean_shift"]<0)).sum())
    sig_df = df[df[sigcol]==True]
    summary.append({
        "Condition":              label,
        "Total Codes":            f"{total:,}",
```

```
        "Significant (FDR)":       f"{n_sig:,}",
        "% Significant":           f"{100*n_sig/total:.1f}%",
        "Upward ↑":                f"{up:,}",
        "Downward ↓":              f"{down:,}",
        "Max Δ Upward":            f"{sig_df['mean_shift'].max():+.5f}" if n_sig else "-",
        "Max Δ Downward":          f"{sig_df['mean_shift'].min():+.5f}" if n_sig else "-",
        "Median |d| (sig only)":   f"{sig_df['cohens_d'].abs().median():.3f}" if n_sig else "-
    })

pd.DataFrame(summary)
```

---

4 Discussion

4.1 Pejorative Shift: What a Significant Finding Means (RQ1)

If the permutation tests confirm H for a substantial number of codes under pejorative language, the conclusion is direct: **CORe encodes pejorative patient descriptors as diagnostic signals.** The model has learned from the MIMIC-III training corpus that notes containing words like *drug-seeking*, *malingering*, or *non-compliant* tend to co-occur with certain ICD-9 codes. That co-occurrence is real in the data — but it is a spurious association. The correlation exists not because those words are clinically informative, but because those words are applied more often to patients who have already received those diagnoses, for social and systemic reasons documented extensively in the literature (Goddu et al., 2018; Himmelstein et al., 2022).

The codes most plausibly affected are in ICD-9 290–319 (psychiatric and behavioral disorders, substance use codes 303–305, depressive disorder 311) and pain codes (338) — precisely the diagnostic categories that pejorative terms like *drug-seeking* and *malingering* semantically anticipate. If those codes appear among the most shifted in **?@tbl-pejorative-top**, this hypothesis is confirmed empirically.

4.2 Laudatory Shift: Suppression Bias (RQ2)

A significant downward shift under laudatory language for the same code categories would confirm **bidirectional valence bias**: positive framing suppresses the very predictions that pejorative framing elevates. This is clinically dangerous in a different way. A patient described as *compliant* or *cooperative* who genuinely has alcohol dependence may have that diagnosis under-predicted, leading to missed clinical coding and potentially missed intervention. Kelly & Westerhoff (2010) documented this pattern in human clinician judgment; a finding of computational replication would mean AI amplifies rather than corrects the human bias.

4.3 Neutral Valence Shift: Separating Structural from Valence Effects (RQ3)

The neutral condition directly answers a confounding question: are RQ1 and RQ2 findings about *valence* or about *any inserted word*?

If neutral-term shifts are comparable in magnitude and count to pejorative and laudatory shifts, then the structural explanation (the model is sensitive to any word inserted at the patient characteristic position) cannot be ruled out. If neutral shifts are substantially smaller, the valence explanation is supported. The comparison of **?@tbl-neutralval-top** with **?@tbl-pejorative-top** and **?@tbl-laudatory-top** is the key evidence for this distinction.

4.4 Asymmetry (RQ4): What the Scatter Plot Reveals

The scatter in **?@fig-asymmetry** and Pearson r directly test whether the model treats valence as a single bipolar axis. An $r \approx -1$ would mean calling a patient "non-compliant" raises code 311 by as much as calling them "compliant" lowers it — simple symmetric bias. An $r \approx 0$ or positive would mean the two poles affect *different* codes or affect the *same* codes in the same direction — a more complex and clinically concerning bias structure where both positive and negative evaluative language push predictions toward the same diagnostic categories.

4.5 Attention Shifts: What They Add (RQ5)

?@fig-attention-shift moves the analysis from *observing* that predictions shift to *explaining how*. The critical visual check is whether the inserted valence term itself appears among the highest-shifted words. If it does, the model is demonstrably routing clinically irrelevant language through its diagnostic pathway — not ignoring it.

This matters for mitigation strategy. If the bias operates through direct high attention on valence tokens, inference-time token masking or adversarial debiasing targeted at those positions is a candidate remedy. If the bias is diffuse — spread across contextual representations of surrounding tokens — a different architectural or training-time intervention is needed. The attention analysis provides this diagnostic information.

The important caveat bears repeating: elevated attention weight on a token is evidence that the model uses it, not proof that it causally drives output (Jain & Wallace, 2019). The probability shift results are primary; attention shifts are supporting mechanistic evidence.

4.6 Fairness and Clinical Deployment Implications

The study's findings connect directly to documented real-world health disparities. Pejorative language is more common in notes about Black patients, patients with chronic pain, and patients with substance use disorder (Himmelstein et al., 2022; Obermeyer et al., 2019; Sun et al., 2022). A model sensitive to this language produces disparate predictions across those groups even without ever receiving race, pain status, or substance use history as explicit features. The model does not need to observe a protected attribute directly to propagate inequity — it only needs to observe language that correlates with it (Char et al., 2020; Chen et al., 2020; Gianfrancesco et al., 2018).

Any deployment of CORe or similar EHR-trained models in clinical coding or decision support should be preceded by this type of behavioral audit. Standard accuracy benchmarks will not detect this bias because the bias is consistent with the training distribution and does not show up as a performance deficit on held-out data.

---

5 Limitations

Single-site data. All notes originate from MIMIC-III (Beth Israel Deaconess Medical Center). Language patterns and their correlations with diagnostic codes may differ across health systems.

Artificial term insertion. The shift design inserts one valence term at a heuristically detected position. Real clinical notes with pejorative language differ in surrounding context, density of evaluative language, and interaction with other note features.

Single attention head. Only layer 11, head 11 is analyzed. This is the production configuration, but other heads may encode complementary patterns.

Attention does not imply causation. High CLS attention on a token does not prove it causally drives predictions (Jain & Wallace, 2019).

Population-level analysis only. Shifts are averaged across all notes. Subgroup analyses — whether effects are larger for notes about patients with documented social disadvantage — are a necessary next step.

Single term per condition. Each note receives one randomly selected term. Results may vary by specific term chosen within a level.

---

6 Conclusion

This study applies systematic behavioral testing to quantify the sensitivity of the CORe clinical ICD-9 coding model to pejorative, laudatory, and neutral valence language across all 1,266 codes. The analysis uses paired probability shifts within each note, tested with paired approximate randomization (10,000 permutations), with effect sizes, bootstrap confidence intervals, and Benjamini-Hochberg FDR correction. Attention weight shifts at layer 11, head 11 provide mechanistic supporting evidence. The cross-condition asymmetry analysis tests whether the two valence poles operate symmetrically or independently.

A model that responds to valence language is a model whose predictions depend on *how* a patient is described rather than *what is clinically true* about them. That is a direct threat to equitable AI-assisted clinical care — and it is not detectable by standard performance benchmarks. Behavioral testing of this kind is a prerequisite for responsible deployment.

---

7 References

Aken, B. van, Papaioannou, J.-M., Mayrdorfer, M., Budde, K., Gers, F. A., & Löser, A. (2021). Clinical outcome prediction from admission notes using self-supervised knowledge integration. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021, Online, April 19 - 23, 2021*.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, *57*(1), 289–300. https://doi.org/10.1111/j.2517-6161.1995.tb02031.x

Char, D. S., Shah, N. H., & Magnus, D. (2020). Implementing machine learning in health care — addressing ethical challenges. *New England Journal of Medicine*, *378*(11), 981–983. https://doi.org/10.1056/NEJMp1714229

Chen, I. Y., Joshi, S., & Ghassemi, M. (2020). Treating health disparities with artificial intelligence. *Nature Medicine*, *26*(1), 16–17. https://doi.org/10.1038/s41591-019-0649-2

Cohen, J. (1988). *Statistical power analysis for the behavioral sciences*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT 2019*, 4171–4186. https://doi.org/10.18653/v1/N19-1423

Efron, B., & Tibshirani, R. J. (1994). *An introduction to the bootstrap*. Chapman & Hall/CRC. https://doi.org/10.1201/9780429246593

Gianfrancesco, M. A., Tamang, S., Yazdany, J., & Schmajuk, G. (2018). Potential biases in machine learning algorithms using electronic health record data. *JAMA Internal Medicine*, *178*(11), 1544–1547. https://doi.org/10.1001/jamainternmed.2018.3763

Goddu, A. P., O'Conor, K. J., Lanzkron, S., Saheed, M. O., Saha, S., Peek, M. E., Haywood, C., & Beach, M. C. (2018). Do words matter? Stigmatizing language and the transmission of bias in the medical record. *Journal of General Internal Medicine*, *33*(5), 685–691. https://doi.org/10.1007/s11606-017-4289-2

Goldberger, A. L., Amaral, L. A. N., Glass, L., Hausdorff, J. M., Ivanov, P. Ch., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C.-K., & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals. *Circulation*, *101*(23), e215–e220. https://doi.org/10.1161/01.CIR.101.23.e215

Hedges, L. V. (1981). Distribution theory for glass's estimator of effect size and related estimators. *Journal of Educational Statistics*, *6*(2), 107–128. https://doi.org/10.3102/10769986006002107

Heider, P. M., & Obeid, J. S. (2023). Approximate randomization testing for natural language processing model evaluation in clinical informatics. *Journal of the American Medical Informatics Association*, *30*(6), 1062–1070. https://doi.org/10.1093/jamia/ocad040

Himmelstein, G., Bates, D., & Zhou, L. (2022). Examination of stigmatizing language in the electronic health record. *JAMA Network Open*, *5*(1), e2144967. https://doi.org/10.1001/jamanetworkopen.2021.44967

Jain, S., & Wallace, B. C. (2019). Attention is not explanation. *Proceedings of NAACL-HLT 2019*, 3543–3556. https://doi.org/10.18653/v1/N19-1357

Johnson, A. E. W., Pollard, T. J., Shen, L., Lehman, L.-W. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., & Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data*, *3*, 160035. https://doi.org/10.1038/sdata.2016.35

Kelly, J. F., & Westerhoff, C. M. (2010). Does it matter how we refer to individuals with substance-related conditions? A randomized study of two commonly used terms. *International Journal of Drug Policy*, *21*(3), 202–207. https://doi.org/10.1016/j.drugpo.2009.10.010

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*(4), 1234–1240. https://doi.org/10.1093/bioinformatics/btz682

Obermeyer, Z., Powers, B., Vogeli, C., & Mullainathan, S. (2019). Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, *366*(6464), 447–453. https://doi.org/10.1126/science.aax2342

Sun, M., Oliwa, T., Peek, M. E., & Tung, E. L. (2022). Negative patient descriptors: Documenting racial bias in the electronic health record. *Health Affairs*, *41*(2), 203–211. https://doi.org/10.1377/hlthaff.2021.01423

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, *1*(6), 80–83. https://doi.org/10.2307/3001968

Yeh, A. (2000). More accurate tests for the statistical significance of result differences. *Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000)*, 947–953. https://doi.org/10.3115/992730.992783

---

8 Appendix

8.1 A — ICD-9 Chapters Referenced

Range	Chapter
001–139	Infectious and Parasitic Diseases
140–239	Neoplasms
240–279	Endocrine, Nutritional, Metabolic
280–289	Blood and Blood-Forming Organs
290–319	**Mental Disorders (primary RQ1 target)**
320–389	Nervous System and Sense Organs
390–459	Circulatory System
460–519	Respiratory System
520–579	Digestive System
580–629	Genitourinary System
800–999	Injury and Poisoning
V codes	Supplementary Health Factors
E codes	External Causes of Injury

8.2 B — Statistical Decision Logic (Annotated)

```
For each ICD-9 code k (k = 1 … 1,266):


    PRIMARY QUESTION: Is the mean shift real or due to chance?

    Paired permutation test (T = 10,000 sign-flip permutations)
```

```
    H : mean( _ik) = 0
    p = (#|perm mean|   |obs mean| + 1) / (T + 1)

      Two-sided     Non-parametric     No normality assumption

                              ↓
                Benjamini-Hochberg FDR correction
                (1,266 simultaneous tests, q = 0.05)
                              ↓

           Significant (q<0.05)   Not significant
          ↓                                    → Conclude: no
                         evidence of valence
    SECONDARY QUESTION:              effect on this code
    How large is the effect?

    Cohen's d = ¯ / SD( )
    Hedges' g  (bias-corrected)
    Bootstrap 95% CI on ¯

          ↓
  Effect size thresholds:
    |d| < 0.2  → negligible
    |d| < 0.5  → small
    |d| < 0.8  → medium
    |d|  0.8  → large

  Supplementary tests (parallel, not primary):
    Paired t-test        - parametric (assumes normality of  )
    Wilcoxon signed-rank - tests median, robust to outliers
    Both FDR-corrected. Agreement with permutation = higher confidence.
```

8.3 C — Attention Weight Pipeline (Annotated)

```
Raw output from CORe for each note:
  attentions[layer=11][head=11]  →  L × L matrix
  CLS row = attentions[11][11][0, 1:L-1]   ← exclude [CLS] and [SEP] tokens

Sub-token → word aggregation (average method, prevents sub-token bias):
  For each original word:
    collect all sub-token CLS weights
    word_weight = mean(sub-token weights)
  Renormalize: word_weights /= sum(word_weights)
  Result: one weight per word, renormalized to sum to 1

Statistical analysis of attention shifts:
```

```
  For each word w in both baseline and shift CSVs:
    1. Pivot: rows = NoteID, column = word, cell = attention weight
       (fill with 0 when word absent from a note)
    2. Compute Δ_w = mean(shift_weight − baseline_weight) across notes
    3. Paired permutation test on the n attention differences → p-value
    4. Benjamini-Hochberg FDR across all words tested

Key output:
  Words with Δ_w > 0 and significant p → model attends MORE to that
  word under the shift condition.
  If the inserted valence term appears here → model uses it as a signal.
```

8.4 D — Running the Full Pipeline

```
# Step 1: Generate prediction and attention CSVs for all four conditions
python main.py \
  --test_set_path ./data/DIA_GROUPS_3_DIGITS_adm_test.csv \
  --model_path DATEXIS/CORe-clinical-diagnosis-prediction \
  --shift_keys neutralize,pejorative,laud,neutralval \
  --task diagnosis \
  --save_dir ./results \
  --gpu true \
  --batch_size 768 \
  --random_seed 42

# Step 2: Run comprehensive statistical analysis
python analyze_valence_results.py \
  --results_dir ./results \
  --baseline_key neutralize \
  --conditions pejorative,laud,neutralval \
  --n_permutations 10000 \
  --alpha 0.05 \
  --correction fdr_bh \
  --seed 42 \
  --output_dir ./results/analysis

# Step 3: Render this report
quarto render valence_behavioral_testing.qmd --to html
quarto render valence_behavioral_testing.qmd --to pdf
```