Deconstructing the Failure of Ideal Noise Correction: A Three-Pillar Diagnosis

Supplementary Material

A. Proofs for the Ideal Fitted Case (Theorem 4.2)

In this appendix, we provide the rigorous derivation for the performance of Forward Correction (FC) and No Correction (NC) in the ideal asymptotic limit ($N \to \infty$). We rely on the notation defined in the main text: $\eta(x) \triangleq P(Y|X=x)$ denotes the clean posterior, and $\eta^n(x) \triangleq P(Y^n|X=x)$ denotes the noisy posterior. Recall the definition of inherent uncertainty: $\delta(x) \triangleq 1 - \max_k \eta_k(x)$.

A.1. Proof of Theorem 4.2(a): Accuracy Analysis

Proof. We analyze the expected accuracy for both methods and derive the performance gap Δ.

1. Forward Correction (FC). Under the ideal fitted assumption, FC is statistically consistent. It successfully recovers the clean posterior, i.e., $\hat{p}_{\texttt{FC}}(x) = \eta(x)$. Consequently, the model's prediction $Y^f_{\texttt{FC}}(x)$ coincides with the Clean Bayes-Optimal classifier $Y^*(x) = \arg\max_k \eta_k(x)$. The accuracy is derived as:

$$\begin{aligned}
\mathrm{ACC}(f_{\texttt{FC}}) &= \mathbb{E}_{(X,Y)}\left[\mathbb{I}(Y = Y^f_{\texttt{FC}}(X))\right] \\
&= \mathbb{E}_X\left[\sum_{y\in\mathcal{Y}} P(Y=y|X)\cdot\mathbb{I}(y=Y^*(X))\right] \\
&= \mathbb{E}_X\left[\eta_{Y^*(X)}(X)\right] \\
&= \mathbb{E}_X\left[1-\delta(X)\right] = 1 - \mathbb{E}_X[\delta(X)].
\end{aligned} \tag{8}$$

This confirms the first claim of Theorem 4.2(a).

2. No Correction (NC). The NC objective minimizes the risk with respect to the noisy labels. The population minimizer yields the noisy posterior $\hat{p}_{\texttt{NC}}(x) = \eta^n(x)$. Thus, the prediction follows the Noisy Bayes-Optimal classifier $\tilde{Y}^*(x) = \arg\max_k \eta^n_k(x)$. Using the partition from Definition 4.1, we decompose the accuracy via the Law of Total Expectation:

$$\begin{aligned}
\mathrm{ACC}(f_{\texttt{NC}}) &= \mathbb{E}_X\left[P(Y=\tilde{Y}^*(X)\mid X)\right] \\
&= P(\mathcal{X}_{\text{correct}})\cdot\mathbb{E}_{X|\mathcal{X}_{\text{correct}}}\left[\eta_{\tilde{Y}^*(X)}(X)\right] + P(\mathcal{X}_{\text{error}})\cdot\mathbb{E}_{X|\mathcal{X}_{\text{error}}}\left[\eta_{\tilde{Y}^*(X)}(X)\right].
\end{aligned} \tag{9}$$

We analyze the term $\eta_{\tilde{Y}^*(X)}(X)$ in each regime:

- **Regime 1 ($\mathcal{X}_{\text{correct}}$):** By definition, $\tilde{Y}^*(x) = Y^*(x)$. Thus, $\eta_{\tilde{Y}^*(x)}(x) = \eta_{Y^*(x)}(x) = 1-\delta(x)$.
- **Regime 2 ($\mathcal{X}_{\text{error}}$):** By definition, the predicted class is strictly suboptimal, i.e., $\tilde{Y}^*(x) = k_{err}$ where $k_{err} \neq Y^*(x)$.

3. The Accuracy Gap (Δ). The gap is defined as $\Delta \triangleq \mathrm{ACC}(f_{\texttt{FC}}) - \mathrm{ACC}(f_{\texttt{NC}})$. Decomposing $\mathrm{ACC}(f_{\texttt{FC}})$ similarly over the partition, the terms on $\mathcal{X}_{\text{correct}}$ are identical and cancel out. We are left with the difference on the error set:

$$\Delta = P(\mathcal{X}_{\text{error}})\cdot\mathbb{E}_{X|\mathcal{X}_{\text{error}}}\left[\eta_{Y^*(X)}(X) - \eta_{\tilde{Y}^*(X)}(X)\right]. \tag{10}$$

For any $x \in \mathcal{X}_{\text{error}}$, let $k_{err} = \tilde{Y}^*(x)$. The probability mass of this erroneous class, $\eta_{k_{err}}(x)$, is bounded by two constraints:
1. It is sub-optimal: $\eta_{k_{err}}(x) \le \eta_{Y^*(x)}(x) = 1-\delta(x)$.
2. It is part of the residual mass: $\eta_{k_{err}}(x) \le \sum_{k\neq Y^*}\eta_k(x) = \delta(x)$.

Thus, $\eta_{k_{err}}(x) \le \min(\delta(x), 1-\delta(x))$. Substituting this into the gap equation:

$$\begin{aligned}
\eta_{Y^*(x)}(x) - \eta_{k_{err}}(x) &\ge (1-\delta(x)) - \min(\delta(x), 1-\delta(x)) \\
&= \max(0, 1-2\delta(x)).
\end{aligned} \tag{11}$$

Therefore, the lower bound for the gap is:

$$\Delta \geq P(\mathcal{X}_{\text{error}}) \cdot \mathbb{E}_{X|\mathcal{X}_{\text{error}}}\Big[\max\big(0, 1-2\delta(X)\big)\Big] \geq 0. \tag{12}$$

This proves the non-negative gap and completes the proof for Part (a). □

A.2. Proof of Theorem 4.2(b): ECE Analysis

Proof. Let $C(X) = \max_k \hat{p}_k(X)$ be the prediction confidence and $Y^f(X) = \arg\max_k \hat{p}_k(X)$ be the predicted label. The Expected Calibration Error (ECE) is defined as:

$$\text{ECE}(f) = \mathbb{E}_C\left[\left|P(Y = Y^f(X) \mid C(X) = C) - C\right|\right]. \tag{13}$$

By defining the *per-sample calibration gap* as $\Delta_{\text{cal}}(x) \triangleq |P(Y = Y^f(x)|x) - \hat{p}_{Y^f(x)}(x)|$, we can rewrite the ECE as the expected local calibration error:

$$\text{ECE}(f) = \mathbb{E}_X\left[\Delta_{\text{cal}}(X)\right]. \tag{14}$$

1. Forward Correction (FC). In the ideal case, $\hat{p}_{\texttt{FC}}(x) = \eta(x)$. The confidence is $C(x) = \eta_{Y^*(x)}(x)$. The true accuracy of this prediction is $P(Y = Y^*(x)|x) = \eta_{Y^*(x)}(x)$. Since the model output matches the ground truth posterior, the gap vanishes:

$$\Delta_{\text{cal}}^{\texttt{FC}}(x) = |\eta_{Y^*(x)}(x) - \eta_{Y^*(x)}(x)| = 0 \implies \text{ECE}(f_{\texttt{FC}}) = 0. \tag{15}$$

2. No Correction (NC). In the ideal case, $\hat{p}_{\texttt{NC}}(x) = \eta^n(x)$. The model's confidence is derived from the noisy posterior: $C(x) = \eta^n_{\tilde{Y}^*(x)}(x)$. However, the true correctness of the prediction depends on the clean posterior: $P(Y = \tilde{Y}^*(x)|x) = \eta_{\tilde{Y}^*(x)}(x)$. The calibration gap is:

$$\Delta_{\text{cal}}^{\texttt{NC}}(x) = \left|\eta_{\tilde{Y}^*(x)}(x) - \eta^n_{\tilde{Y}^*(x)}(x)\right|. \tag{16}$$

Unless the transition matrix $T(x)$ is the identity (no noise) or a specific permutation that preserves diagonal dominance exactly, generally $\eta(x) \neq \eta^n(x)$. Thus, $\Delta_{\text{cal}}^{\texttt{NC}}(x) > 0$ for some x, implying $\text{ECE}(f_{\texttt{NC}}) > 0$. □

B. Derivation of the Empirical Minimizer for FC (Eq. 6)

In this section, we provide the formal derivation for the optimal solution $\hat{p}(x)$ that minimizes the single-sample Forward Correction (FC) loss.

1. Optimization Problem. We analyze the failure mode where a high-capacity network memorizes the training set, driving the empirical risk $\widehat{R}(f) \to 0$. This implies minimizing the loss for each sample (x, y^n) independently.

The single-sample FC loss for a prediction $\hat{p}(x)$ is:

$$\ell_{\texttt{FC}}(\hat{p}(x) \mid x, y^n) = -\log\left(\sum_{k=1}^{K} T_{k,y^n}(x) \cdot \hat{p}_k(x)\right). \tag{17}$$

Our objective is to find the optimal probability vector $\hat{p}^*$ that minimizes this loss, subject to the constraint that $\hat{p}^*$ lies on the probability simplex Δ^{K-1}:

$$\hat{p}^* = \arg\min_{\hat{p} \in \Delta^{K-1}} \ell_{\texttt{FC}}(\hat{p} \mid x, y^n). \tag{18}$$

2. Equivalent Linear Objective. The function $g(z) = -\log(z)$ is strictly monotonically decreasing for $z > 0$. Therefore, minimizing $\ell_{\texttt{FC}}$ is mathematically equivalent to maximizing its argument:

$$\hat{p}^* = \arg\max_{\hat{p} \in \Delta^{K-1}} \left(\sum_{k=1}^{K} T_{k,y^n}(x) \cdot \hat{p}_k\right). \tag{19}$$

3. Solution via Linear Programming. For a fixed sample (x, y^n), the transition probabilities $T_{k,y^n}(x)$ (which form the y^n-th column of $T(x)$) are constants. Let us define a constant vector $\mathbf{c} \in \mathbb{R}^K$ where each element $c_k = T_{k,y^n}(x)$.

The optimization problem simplifies to:

$$\arg \max_{\hat{p} \in \Delta^{K-1}} \left(\mathbf{c}^\top \hat{p} \right). \tag{20}$$

This is a **Linear Program (LP)**: we are maximizing a linear objective function ($\mathbf{c}^\top \hat{p}$) over a convex polytope (the probability simplex Δ^{K-1}).

4. Optimal Solution at Vertex. A fundamental theorem of linear programming states that the maximum of a linear function over a convex polytope must be achieved at one of the polytope's vertices. The vertices of the probability simplex Δ^{K-1} are the set of standard basis vectors (one-hot vectors):

$$\mathcal{V} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_K\}.$$

To find the optimal solution, we evaluate the objective function at each vertex $\mathbf{e}_j$:

$$\mathbf{c}^\top \mathbf{e}_j = \sum_{k=1}^{K} c_k \cdot (\mathbf{e}_j)_k = c_j = T_{j,y^n}(x).$$

The objective is maximized by choosing the vertex $\mathbf{e}_{k^*}$ that corresponds to the largest coefficient c_{k^*}. We define this optimal index k^* as:

$$k^*_{\texttt{FC}}(x) \triangleq \arg \max_{k \in \{1,\ldots,K\}} T_{k,y^n}(x).$$

Therefore, the unique optimal solution that minimizes the single-sample FC loss is:

$$\hat{p}^* = \mathbf{e}_{k^*_{\texttt{FC}}(x)}.$$

This completes the derivation of Equation (6).

Implications. This derivation formally proves that the empirical minimizer of the FC loss is *not* the "correct" soft posterior $\eta(x)$. Instead, the loss function creates an optimization landscape whose global minimum (for a finite sample) is a hard, one-hot vector.

C. Proofs for the Empirical Overfitted Case (Theorem 4.3)

In this appendix, we provide the detailed derivations for the accuracy and calibration properties in the "Empirical Overfitted Case" (Section 4.2.2). We assume the model has memorized the training data, collapsing its predictions to one-hot vectors as derived in Section B.

We use the notation $\eta_k(x) \triangleq P(Y = k \mid X = x)$ for the clean posterior and $\delta(x) \triangleq 1 - \eta_{Y^*}(x)$ for the inherent uncertainty. The accuracy is $\mathrm{ACC} = \mathbb{E}_X[P(Y^f = Y \mid X)]$.

C.1. Proof of Theorem 4.3(a): Accuracy (ACC)

We first derive the exact conditional accuracy $\mathrm{ACC}(x) = P(Y^f = Y \mid X = x)$ for both methods.

No Correction (NC) The NC solution memorizes the noisy label, so $Y^f = Y^n$. The conditional accuracy is the probability that the observed noisy label matches the (unseen) true clean label.

$$\begin{aligned}
\mathrm{ACC}_{\texttt{NC}}(x) &= P(Y^f = Y \mid x) = P(Y^n = Y \mid x) \\
&= \sum_{k \in \mathcal{Y}} P(Y^n = Y, Y = k \mid x) \\
&= \sum_{k \in \mathcal{Y}} P(Y^n = k, Y = k \mid x) \quad \text{(since } Y^n = Y \text{ implies } Y^n = k \text{ and } Y = k\text{)} \\
&= \sum_{k=1}^{K} P(Y^n = k \mid Y = k, x) P(Y = k \mid x) \\
&= \sum_{k=1}^{K} T_{k,k}(x) \eta_k(x)
\end{aligned} \tag{21}$$

To analyze its dependence on $\delta(x)$, we split the sum into the dominant class ($k = Y^*$) and the residual $\mathcal{R}_{\texttt{NC}}(x)$:

$$\begin{aligned}
\mathrm{ACC}_{\texttt{NC}}(x) &= \eta_{Y^*}(x) T_{Y^*,Y^*}(x) + \sum_{k \neq Y^*} \eta_k(x) T_{k,k}(x) \\
&= (1 - \delta(x)) T_{Y^*,Y^*}(x) + \mathcal{R}_{\texttt{NC}}(x)
\end{aligned} \tag{22}$$

Since $0 \leq T_{k,k}(x) \leq 1$ and $\sum_{k \neq Y^*} \eta_k(x) = \delta(x)$, the residual is bounded: $0 \leq \mathcal{R}_{\texttt{NC}}(x) \leq \delta(x)$. Thus, $\mathcal{R}_{\texttt{NC}}(x) = \mathcal{O}(\delta(x))$. The dominant first-order approximation is:

$$\mathrm{ACC}_{\texttt{NC}}(x) \approx (1 - \delta(x)) T_{Y^*,Y^*}(x) \tag{23}$$

Forward Correction (FC) From Section B, the FC prediction is $Y^f = k^*(Y^n)$, where $k^*(j) \triangleq \arg\max_k T_{k,j}(x)$. The conditional accuracy is the probability that this prediction matches the true label Y.

$$\begin{aligned}
\mathrm{ACC}_{\texttt{FC}}(x) &= P(Y^f = Y \mid x) = P(k^*(Y^n) = Y \mid x) \\
&= \sum_{k \in \mathcal{Y}} P(k^*(Y^n) = Y, Y = k \mid x) \\
&= \sum_{k \in \mathcal{Y}} P(Y = k \mid x) P(k^*(Y^n) = k \mid Y = k, x) \quad \text{(as } Y^n \text{ depends on } Y\text{)} \\
&= \sum_{k=1}^{K} \eta_k(x) \left(\sum_{j : k^*(j) = k} P(Y^n = j \mid Y = k, x) \right) \\
&= \sum_{k=1}^{K} \eta_k(x) \underbrace{\left(\sum_{j : k^*(j) = k} T_{k,j}(x) \right)}_{\triangleq C_k(x)}
\end{aligned} \tag{24}$$

Let $C_k(x)$ be the sum of probabilities of transitions from class k to any noisy label j that maps back to k. The exact accuracy is $\mathrm{ACC}_{\texttt{FC}}(x) = \sum_k \eta_k(x) C_k(x)$. We split this sum:

$$\begin{aligned}
\mathrm{ACC}_{\texttt{FC}}(x) &= \eta_{Y^*}(x) C_{Y^*}(x) + \sum_{k \neq Y^*} \eta_k(x) C_k(x) \\
&= (1 - \delta(x)) C_{Y^*}(x) + \mathcal{R}_{\texttt{FC}}(x)
\end{aligned} \tag{25}$$

Since $C_k(x) = \sum_{j:\ldots} T_{k,j}(x) \leq \sum_j T_{k,j}(x) = 1$, the residual $\mathcal{R}_{\texttt{FC}}(x)$ is also $\mathcal{O}(\delta(x))$. The first-order approximation is:

$$\mathrm{ACC}_{\texttt{FC}}(x) \approx (1 - \delta(x)) C_{Y^*}(x) \tag{26}$$

Comparison: The Gain/Loss Trade-off The relative performance is driven by the difference between $C_{Y^*}(x)$ and $T_{Y^*,Y^*}(x)$. The first-order accuracy gap is:

$$\Delta_{\text{ACC}}(x) \approx (1 - \delta(x)) \left[C_{Y^*}(x) - T_{Y^*,Y^*}(x) \right] \tag{27}$$

To understand this gap, we decompose $C_{Y^*}(x)$ by splitting its sum at $j = Y^*$:

$$\begin{aligned} C_{Y^*}(x) &= \sum_{j:k^*(j)=Y^*} T_{Y^*,j}(x) \\ &= T_{Y^*,Y^*}(x) \cdot \mathbb{I}(k^*(Y^*) = Y^*) + \sum_{j \neq Y^*} T_{Y^*,j}(x) \cdot \mathbb{I}(k^*(j) = Y^*) \end{aligned} \tag{28}$$

Substituting this into the gap equation $[C_{Y^*}(x) - T_{Y^*,Y^*}(x)]$ yields:

$$\begin{aligned} &= \left[T_{Y^*,Y^*}(x)\mathbb{I}(k^*(Y^*) = Y^*) + \sum_{j \neq Y^*} \dots \right] - T_{Y^*,Y^*}(x) \\ &= \sum_{j \neq Y^*} T_{Y^*,j}(x)\mathbb{I}(k^*(j) = Y^*) + T_{Y^*,Y^*}(x) \left(\mathbb{I}(k^*(Y^*) = Y^*) - 1 \right) \\ &= \underbrace{\sum_{j \neq Y^*} T_{Y^*,j}(x)\mathbb{I}(k^*(j) = Y^*)}_{\text{Gain: Errors corrected by FC}} - \underbrace{T_{Y^*,Y^*}(x)\mathbb{I}(k^*(Y^*) \neq Y^*)}_{\text{Loss: Correct labels mis-corrected by FC}} \end{aligned} \tag{29}$$

This confirms that FC's performance is a fragile trade-off: it gains accuracy only if it correctly maps erroneous labels (like $j \neq Y^*$) back to Y^*, but it loses accuracy if it maps the *correct* label Y^* to something else ($k^*(Y^*) \neq Y^*$).

Intuition and Visualization of the Gain/Loss Trade-off. To build a strong intuition for the trade-off just derived in Equation (29), we provide a concrete visualization of the collapsed solution $k^*(j) \triangleq \arg\max_k T_{k,j}(x)$. This is a deterministic lookup table found by scanning each **column** j of the T-matrix to find the **row** k with the maximum value.

Consider this 3-class "pathological" but valid T-matrix (rows sum to 1, column max bolded):

$$T(x) = \begin{array}{r} \\ Y = \text{Bird (k=1)} \\ Y = \text{Dog (k=2)} \\ Y = \text{Cat (k=3)} \end{array} \begin{array}{c} \begin{array}{ccc} Y^n = \text{Bird (j=1)} & Y^n = \text{Dog (j=2)} & Y^n = \text{Cat (j=3)} \end{array} \\ \begin{pmatrix} \mathbf{0.5} & \mathbf{0.45} & 0.05 \\ 0.25 & 0.4 & \mathbf{0.35} \\ 0.33 & 0.33 & 0.34 \end{pmatrix} \end{array}$$

The lookup table $k^*(j)$ becomes (based on the column-wise bolded maximums):

- $k^*(1) = \arg\max(0.5, 0.25, 0.33) = 1$ (Observed 'Bird' $\rightarrow$ Predict 'Bird')
- $k^*(2) = \arg\max(0.45, 0.4, 0.33) = 1$ (Observed 'Dog' $\rightarrow$ Predict 'Bird')
- $k^*(3) = \arg\max(0.05, 0.35, 0.34) = 2$ (Observed 'Cat' $\rightarrow$ Predict 'Dog')

This pathological mapping $j \rightarrow k^*(j)$ directly explains the Gain/Loss terms from Equation (29):

- **Loss Term Activation ($\mathbb{I}(k^*(Y^*) \neq Y^*)$):** This term activates when a "correct" noisy label is "mis-corrected". For example, if the true label is $Y^* = 2$ ('Dog') and the noisy label is also $Y^n = 2$ (a correct pair), the NC baseline would be correct. However, the FC model uses its lookup table, finds $k^*(2) = 1$, and predicts 'Bird', thus **creating a loss**. The same occurs for $Y^* = 3$ ('Cat'), where the correct label $Y^n = 3$ is mapped to $k^*(3) = 2$.
- **Gain Term Activation ($\mathbb{I}(k^*(j) = Y^*)$ for $j \neq Y^*$):** This term activates when an "incorrect" noisy label is "corrected". For example, if the true label is $Y^* = 1$ ('Bird') but the noisy label is $Y^n = 2$ ('Dog') (an error pair), the NC baseline would be wrong. However, the FC model finds $k^*(2) = 1$, which matches $Y^* = 1$, thus **creating a gain**. The same occurs for $Y^* = 2$ ('Dog') when $Y^n = 3$ ('Cat'), as $k^*(3) = 2$.

Therefore, the performance of the overfitted FC model depends on the fragile balance between these gain and loss terms, which are dictated entirely by the T-matrix structure.

Symmetric Noise Case. Now we apply this framework to the symmetric noise case. With rate $\rho < (K-1)/K$, the T-matrix is constant, and $T_{j,j} = 1-\rho > T_{i,j} = \rho/(K-1)$ for $i \neq j$. This means the maximum of every column j is strictly on the diagonal. Therefore:

$$k^*(j) = j \quad \text{for all } j.$$

We plug this into the Gain/Loss equation (Equation (29)):
- **Gain Term:** $\sum_{j \neq Y^*} T_{Y^*,j} \mathbb{I}(j = Y^*) = 0$ (the sum is over an empty set).
- **Loss Term:** $T_{Y^*,Y^*} \mathbb{I}(Y^* \neq Y^*) = 0$.

The first-order accuracy gap is zero: $\mathrm{ACC}_{\texttt{FC}}(x) \approx \mathrm{ACC}_{\texttt{NC}}(x)$. The total accuracy for both methods collapses to:

$$\mathrm{ACC} \approx \mathbb{E}_X\left[(1-\delta(x))(1-\rho)\right] = (1-\rho)(1-\mathbb{E}[\delta(X)])$$

This analytically confirms the "solution collapse" to the same suboptimal baseline observed in Figure 1.

C.2. Proof of Theorem 4.3(b): ECE

Proof. For both NC and FC, the overfitted solution is a one-hot vector (e.g., $\hat{p} = \mathbf{e}_{k^*}$). The prediction confidence is therefore $C(X) = \max_k \hat{p}_k(X) = 1$ for all samples. The ECE formula is defined as $\mathrm{ECE}(f) = \mathbb{E}_C\left[\left|P(Y = Y^f \mid C) - C\right|\right]$. Since $C = 1$ everywhere, the expectation simplifies to a single point:

$$\begin{aligned}
\mathrm{ECE}(f) &= \left|P(Y = Y^f \mid C = 1) - 1\right| \\
&= \left|P(Y = Y^f) - 1\right| \quad \text{(since } C = 1 \text{ provides no new information)} \\
&= \left|\mathrm{ACC}(f) - 1\right| \\
&= 1 - \mathrm{ACC}(f) \quad \text{(since } \mathrm{ACC}(f) \leq 1\text{)}
\end{aligned} \tag{30}$$

This proves that in the overfitted, hard-label regime, ECE is perfectly and negatively coupled with accuracy. □

D. Gradient Derivation for Forward Correction (FC)

In this section, we derive the gradient of the Forward Corrected loss $\ell_{\texttt{FC}}$ with respect to a single logit $f_k(x)$, as referenced in Section 4.3.

The loss for a single sample (x, y^n) is defined as:

$$\ell_{\texttt{FC}} = -\log(z_{y^n}) \tag{31}$$

where z is the model's predicted noisy posterior vector, $z = T(x)^\top \hat{p}(x)$, and $\hat{p}(x) = \mathrm{softmax}(f(x))$. The term z_{y^n} is the single component corresponding to the observed noisy label y^n:

$$z_{y^n} = \sum_{i=1}^{K} T_{i,y^n}(x)\hat{p}_i(x)$$

We use the chain rule to compute $\frac{\partial \ell_{\texttt{FC}}}{\partial f_k}$:

$$\frac{\partial \ell_{\texttt{FC}}}{\partial f_k} = \frac{\partial \ell_{\texttt{FC}}}{\partial z_{y^n}} \cdot \frac{\partial z_{y^n}}{\partial f_k}$$

Step 1: Derivative of Loss w.r.t. z_{y^n} The derivative of the negative logarithm is:

$$\frac{\partial \ell_{\texttt{FC}}}{\partial z_{y^n}} = -\frac{1}{z_{y^n}}$$

Step 2: Derivative of z_{y^n} w.r.t. logit f_k We apply the chain rule again, summing over all clean probabilities $\hat{p}_j$:

$$\begin{aligned}
\frac{\partial z_{y^n}}{\partial f_k} &= \sum_{j=1}^{K} \frac{\partial z_{y^n}}{\partial \hat{p}_j} \cdot \frac{\partial \hat{p}_j}{\partial f_k} \\
&= \sum_{j=1}^{K} T_{j,y^n}(x) \cdot \frac{\partial \hat{p}_j}{\partial f_k}
\end{aligned}$$

We use the standard softmax derivative $\frac{\partial \hat{p}_j}{\partial f_k} = \hat{p}_j(\delta_{jk} - \hat{p}_k)$, where δ_{jk} is the Kronecker delta.

$$\begin{aligned}
\frac{\partial z_{y^n}}{\partial f_k} &= \sum_{j=1}^{K} T_{j,y^n} \hat{p}_j (\delta_{jk} - \hat{p}_k) \\
&= \underbrace{T_{k,y^n} \hat{p}_k (1 - \hat{p}_k)}_{\text{Case } j=k} + \underbrace{\sum_{j \neq k} T_{j,y^n} \hat{p}_j (-\hat{p}_k)}_{\text{Case } j \neq k} \\
&= (T_{k,y^n} \hat{p}_k - T_{k,y^n} \hat{p}_k^2) - \hat{p}_k \sum_{j \neq k} T_{j,y^n} \hat{p}_j \\
&= T_{k,y^n} \hat{p}_k - \hat{p}_k \left(T_{k,y^n} \hat{p}_k + \sum_{j \neq k} T_{j,y^n} \hat{p}_j \right) \\
&= T_{k,y^n} \hat{p}_k - \hat{p}_k \left(\sum_{j=1}^{K} T_{j,y^n} \hat{p}_j \right) \\
&= T_{k,y^n} \hat{p}_k - \hat{p}_k z_{y^n}
\end{aligned}$$

Step 3: Combining the Terms We substitute the results from Step 1 and Step 2 back into the main chain rule formula:

$$\begin{aligned}
\frac{\partial \ell_{\texttt{FC}}}{\partial f_k} &= \left(-\frac{1}{z_{y^n}} \right) \cdot (T_{k,y^n} \hat{p}_k - \hat{p}_k z_{y^n}) \\
&= -\frac{T_{k,y^n} \hat{p}_k}{z_{y^n}} + \frac{\hat{p}_k z_{y^n}}{z_{y^n}} \\
&= \hat{p}_k - \frac{T_{k,y^n} \hat{p}_k}{z_{y^n}}
\end{aligned} \tag{32}$$

Recalling the definition $z_{y^n} = \sum_j T_{j,y^n} \hat{p}_j$, we obtain the final form presented in Equation (7):

$$\frac{\partial \ell_{\texttt{FC}}}{\partial f_k(x)} = \hat{p}_k - q_k \tag{33}$$

where $q_k = \frac{T_{k,y^n}(x) \hat{p}_k}{\sum_j T_{j,y^n}(x) \hat{p}_j}$. As noted in the main text, this q_k term is precisely the model's estimated reverse posterior $P(Y = k \mid Y^n = y^n, x)$ via Bayes' rule.

This gradient form $\hat{p}_k - q_k$ should be contrasted with the standard Cross-Entropy gradient, $\hat{p}_k - \mathbb{I}\{y^n = k\}$. Instead of a hard '1' pulling the gradient for the observed class, FC uses a "soft" target q_k distributed over all classes k, which explains the "gradient softening" effect discussed in the main paper.

E. Analysis of Optimization Dynamics and Gradient Saturation

In this appendix, we analyze the optimization dynamics of the Forward Correction (FC) loss. As shown in Appendix B, the theoretical global minimum of the single-sample (overfitted) loss is not the clean label $\mathbf{e}_{y^*}$, but the 'collapsed' solution $\mathbf{e}_{k^*_{\texttt{FC}}}$. Here, we show here that the practical optimization path does not even guarantee convergence to this theoretical minimum.

Our analysis proceeds in two steps:

1. **Global Gradient Flow:** We first analyze the vector field of the gradient $\nabla_{\mathbf{f}} \ell_{\texttt{FC}}$. We confirm that the gradient flow, when viewed globally, does indeed point towards the correct theoretical minimum $\mathbf{e}_{k^*_{\texttt{FC}}}$ (the solution from the Linear Program analysis in Appendix B).
2. **Local Gradient Saturation:** We then show that due to the Softmax parameterization, the gradient magnitude vanishes near *all* simplex vertices. This creates 'dead zones' (local minima) around sub-optimal attractors, most notably the noisy label vertex $\mathbf{e}_{y^n}$.

This analysis demonstrates that while the loss function's *global* landscape points to $\mathbf{e}_{k^*_{\texttt{FC}}}$, its *local* properties trap the optimizer. The final 'pseudo-converged' solution is therefore path-dependent, not guaranteed to be the theoretical optimum, and often defaults to the noisy vertex $\mathbf{e}_{y^n}$ due to early-learning dynamics.

Theoretical Optimum and Global Gradient Flow As derived in Appendix B, the FC loss for a sample (x, y^n) is:

$$\ell_{\texttt{FC}}(\hat{\mathbf{p}}) = -\log\left(\sum_{k=1}^{K} \hat{p}_k T_{k,y^n}\right) \tag{34}$$

Minimizing this is equivalent to the Linear Program $\max_{\hat{\mathbf{p}}\in\Delta^{K-1}} \sum_{k=1}^{K} \hat{p}_k T_{k,y^n}$. The global optimum $\hat{\mathbf{p}}^*$ is the one-hot vector $\mathbf{e}_{k^*_{\texttt{FC}}}$, where $k^*_{\texttt{FC}} = \arg\max_k T_{k,y^n}$.

An analysis of the gradient flow (as visualized on a 3-class simplex example in Fig. 4) confirms this. The vector field of the gradient $\nabla_{\mathbf{f}}\ell_{\texttt{FC}}$ globally points away from all other vertices and towards the single vertex $\mathbf{e}_{k^*_{\texttt{FC}}}$. Theoretically, a gradient descent algorithm with perfect information should converge to $k^*_{\texttt{FC}}$.

Gradient Saturation and Pseudo-Convergence The practical failure arises from the Softmax parameterization $\hat{p}_k = \text{softmax}(f_k)$. The gradient of the loss with respect to the logits $\mathbf{f}$ is:

$$\frac{\partial \ell}{\partial f_k} = \sum_{j=1}^{K} \frac{\partial \ell}{\partial \hat{p}_j} \frac{\partial \hat{p}_j}{\partial f_k} \tag{35}$$

The Softmax derivative $\frac{\partial \hat{p}_j}{\partial f_k} = \hat{p}_j(\delta_{jk} - \hat{p}_k)$ approaches 0 as $\hat{\mathbf{p}}$ approaches *any* vertex $\mathbf{e}_i$. Consequently, the magnitude of the logit gradient vanishes near all vertices:

$$\lim_{\hat{\mathbf{p}}\to\mathbf{e}_i} \|\nabla_{\mathbf{f}}\ell_{\texttt{FC}}\| \approx 0 \quad \text{for any } i \in \{1, \dots, K\} \tag{36}$$

This phenomenon creates "gradient plateaus" or "dead zones" around *every* vertex.

Dynamics Analysis: Trapped in a Local Minimum The existence of these "dead zones" means the final convergence point is path-dependent. In Noisy Label Learning, models famously exhibit an "early learning" phase (fitting simple patterns) followed by a "memorization" phase (fitting noisy labels).

1. The model quickly learns to fit the dominant noisy labels, causing its prediction $\hat{\mathbf{p}}$ to approach the noisy vertex $\mathbf{e}_{y^n}$.
2. As $\hat{\mathbf{p}} \to \mathbf{e}_{y^n}$, the model enters the gradient saturation "dead zone" of this vertex.
3. Although $\mathbf{e}_{y^n}$ is not the global minimum of $\ell_{\texttt{FC}}$ (assuming $k^*_{\texttt{FC}} \neq y^n$), the gradient pointing away from $\mathbf{e}_{y^n}$ and towards the true minimum $\mathbf{e}_{k^*_{\texttt{FC}}}$ becomes infinitesimally small.
4. SGD, with limited step size, fails to escape this local basin of attraction.

This results in *pseudo-convergence*: the model remains "trapped" at the wrong label $\mathbf{e}_{y^n}$. Therefore, the actual overfitted solution observed in practice is not the theoretical optimum $\mathbf{e}_{k^*_{\texttt{FC}}}$, but rather a sub-optimal local minimum $\mathbf{e}_{y^n}$ that acts as a strong attractor.

F. Proof of Theorem 4.4: Fundamental Information Cost

We provide the rigorous proof for Theorem 4.4. We restate the definitions for a fixed input $X = x$:

- $I_{\text{clean}}(x) \triangleq I(M; Y \mid X = x)$
- $I_{\text{noisy}}(x) \triangleq I(M; Y^n \mid X = x)$

Our goal is to prove that $I_{\text{noisy}}(x) \leq I_{\text{clean}}(x)$.

1. Establishing the Conditional Markov Chain The data-generating process described in Section 4.4 forms a conditional Markov chain for any fixed $X = x$:

$$M \;\to\; Y \;\to\; Y^n \quad \text{conditioned on } X = x.$$

This holds because:

1. The hypothesis M (which represents the true data-generating process $\eta_m(x)$) determines the distribution for the clean label Y.
2. The noisy label Y^n is generated based *only* on the clean label Y (via the noise channel $T(x)$), without any direct influence from M.

Therefore, given Y, the noisy label Y^n is conditionally independent of the hypothesis M. This conditional independence implies:

$$I(M; Y^n \mid Y, X = x) = 0. \tag{37}$$

2. Applying the Chain Rule for Mutual Information We decompose the joint mutual information $I(M; Y, Y^n \mid X = x)$ in two different ways using the chain rule:

- **Decomposition 1 (Grouping with Y):**

$$\begin{aligned} I(M; Y, Y^n \mid X = x) &= I(M; Y \mid X = x) + \underbrace{I(M; Y^n \mid Y, X = x)}_{\text{Equals 0 by Equation (37)}} \\ &= I(M; Y \mid X = x) \\ &= I_{\text{clean}}(x) \end{aligned} \tag{38}$$

- **Decomposition 2 (Grouping with Y^n):**

$$\begin{aligned} I(M; Y, Y^n \mid X = x) &= I(M; Y^n \mid X = x) + I(M; Y \mid Y^n, X = x) \\ &= I_{\text{noisy}}(x) + I(M; Y \mid Y^n, X = x) \end{aligned} \tag{39}$$

3. Final Derivation By equating Equation (38) and Equation (39), we have:

$$I_{\text{clean}}(x) = I_{\text{noisy}}(x) + I(M; Y \mid Y^n, X = x)$$

By the non-negativity property of mutual information, the final term must be non-negative:

$$I(M; Y \mid Y^n, X = x) \geq 0$$

Therefore, we conclude:

$$I_{\text{clean}}(x) \geq I_{\text{noisy}}(x) \quad \text{or} \quad \mathbf{I}_{\text{noisy}}(\mathbf{x}) \leq \mathbf{I}_{\text{clean}}(\mathbf{x})$$

Furthermore, the inequality is strict, $I_{\text{noisy}}(x) < I_{\text{clean}}(x)$, if $I(M; Y \mid Y^n, X = x) > 0$. This holds for any non-trivial noise channel $T(x)$ (i.e., not an identity or permutation matrix), as the noisy label Y^n becomes a statistically lossy proxy for the true label Y.

G. Experiment Details

G.1. Dataset Details

CIFAR-10/CIFAR-100 Both datasets consist of 50,000 training images. Following established conventions, we evaluate performance under three standard noise models:

- **Symmetric Noise:** Labels are randomly flipped to any of the other classes with a uniform probability.
- **Asymmetric Noise:** Labels are flipped to mimic real-world mistakes between visually similar categories (e.g., Horse $\leftrightarrow$ Deer and Dog $\leftrightarrow$ Cat).
- **Instance-Dependent Noise (IDN):** To approximate IDN without the prohibitive cost of a unique transition matrix for every sample, we use a grouped setting. The dataset is divided into 50 groups, with each group being assigned a different, randomly generated, diagonally-dominant transition matrix T.

We test a comprehensive range of noise levels: 20%, 50%, 80%, and 90% for symmetric noise, and 40% for asymmetric noise.

Clothing1M Clothing1M [66] is a large-scale, real-world benchmark for LNL. It contains 1 million images in 14 classes, crawled from online shopping websites. The dataset has a substantial level of intrinsic noise, estimated at approximately 38.5%.

G.2. Implementation Details

CIFAR-10/CIFAR-100 We use a PreActResNet-18 [24] backbone for all CIFAR experiments.

- **Standard Training:** The network is trained for 120 epochs using SGD with a weight decay of 5e-4 and a batch size of 128. The initial learning rate is 0.02, decaying to 0.002 at 60 epochs and 0.0002 at 80 epochs.
- **Long Training (for Figure 1):** To observe the long-term collapse, we train for 800 epochs. The learning rate schedule is extended, decaying at 400 and 600 epochs.

Clothing1M Following the setup of [35], we use a ResNet-50 backbone pretrained on ImageNet. The network is trained for 150 epochs with a weight decay of 1e-3 and a batch size of 32. The initial learning rate is 0.002, decaying by a factor of 10 at 50 and 100 epochs.

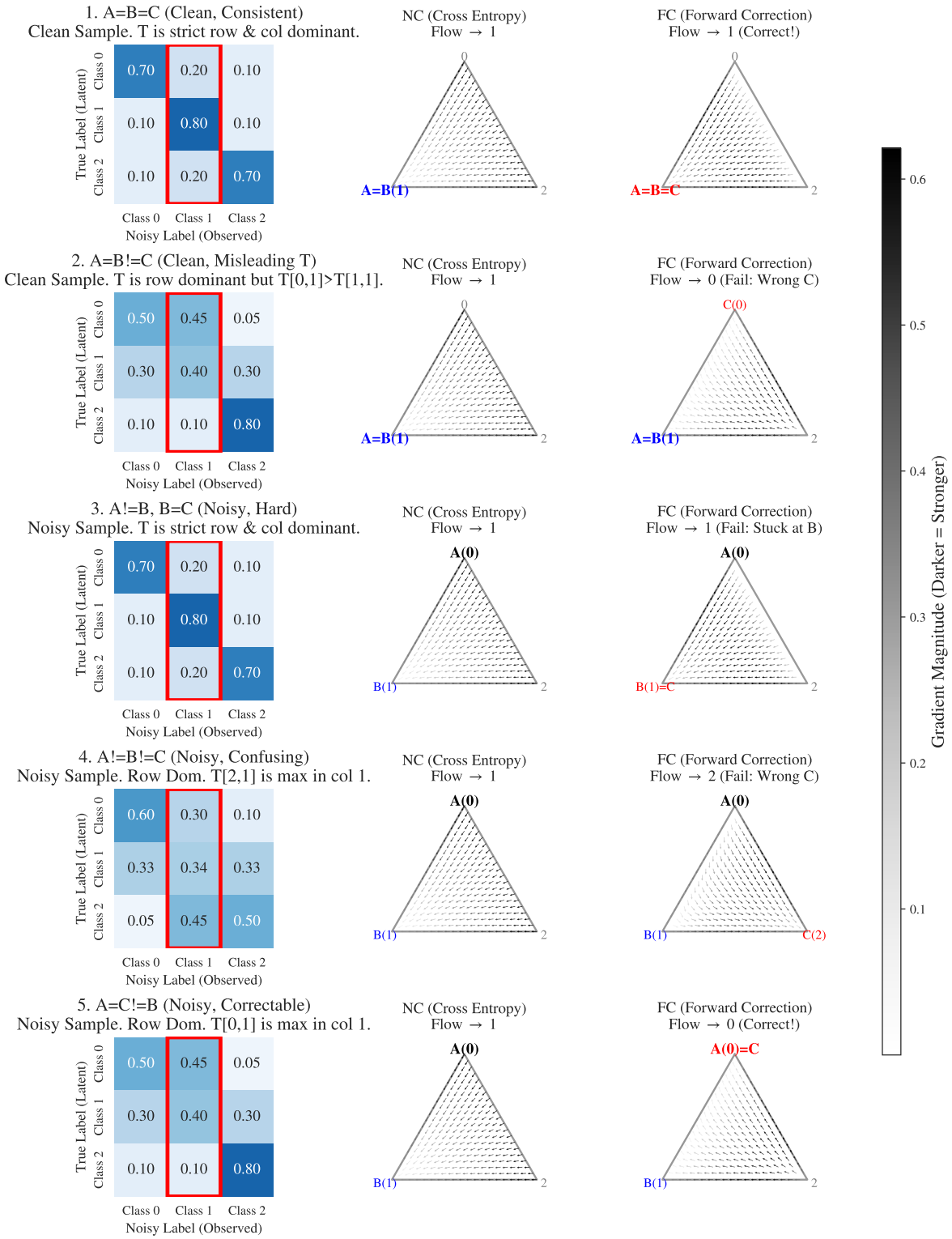

Figure 4. Gradient vector field of the FC loss on a 3-class simplex. We denote the clean label vertex as A ($\mathbf{e}_{y^*}$), the noisy label as B ($\mathbf{e}_{y^n}$), and the theoretical FC optimum as C ($\mathbf{e}_{k^*_{\text{FC}}}$). The vector field confirms that the global minimum is at C. However, the noisy vertex B acts as a strong, non-optimal attractor. The vanishing gradient magnitude ("dead zone") near B traps SGD, leading to the 'pseudo-convergence' analyzed in Section E.