

Identification for Effects, Values of GenAI Content

Ravi B. Sojitra
Stanford University

First draft: [November 25, 2025]

This Draft: [December 1, 2025]

Current Draft: [[Link](#)]

Summary

1. Synthetic content (e.g. GenAI text, image, and video ads) is increasingly used to optimize human behavior (e.g. clicks and purchases) via a two step process.
2. First, several pieces of content are generated, and then a subset with the highest estimated values is shown to people. These estimates often come from vanilla predictive models trained to predict behaviors given content features (e.g. embeddings).
3. My current view is that it is probably better to default to using the G-formula instead of vanilla prediction though the latter can work if the content is not personalized.
4. People experienced in building predictive models will find the G-formula straightforward to implement. While other causal inference and reinforcement learning estimators could work, they introduce considerable implementation complexity.
5. Submit feedback by creating a GitHub Issue at the [public-notes](#) repository. Thanks!

Background: Outcomes are optimized using GenAI content

Today, synthetic content is generated by providing contextual information, instructions, and response desiderata as input to GenAI systems¹. For example, in the figure below, I illustrate an example prompt that I provided as input to a Large Language Model (LLM), which then provided a response. The illustrated prompt contains (1) financial news article contents and the country a user is located in, (2) my request for a good news article title, and (3) a character limit that I want the GenAI system to honor. In this case, the LLM responds with titles written in Spanish because I said the person is located in Spain. Importantly, the semantics, language, and writing style vary based on languages and norms local to the geographic region I provide.

¹Publicly documented examples for marketing: [Angelopoulos et al. \(2024\)](#) and [Batista and Ross \(2024\)](#).

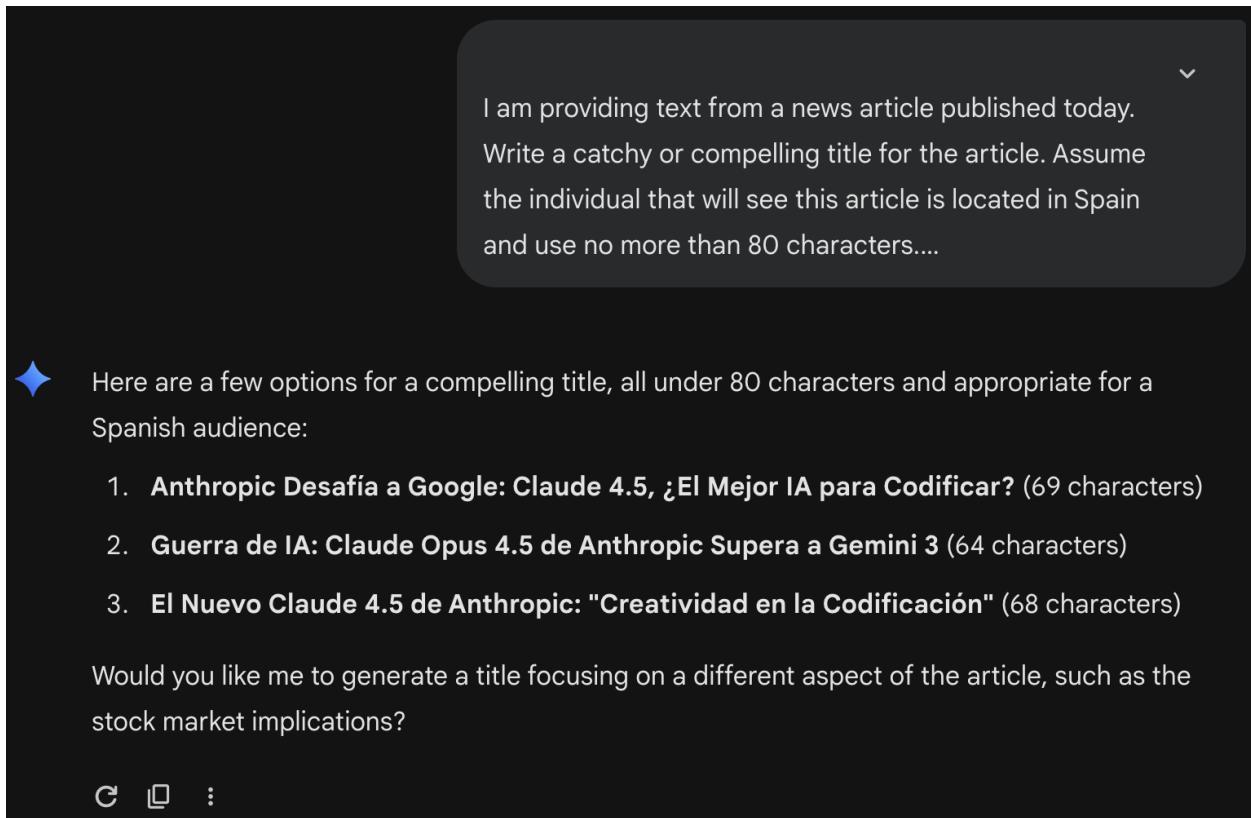


Figure 1: The gray box contains my prompt. Below that, is an LLM response. The original title was “Anthropic launches Claude Opus 4.5 as Google’s Gemini 3 gains big backers”.

After collecting data from many people, a common post-hoc goal is to improve the content shown to people in the future. Two ways this is done is by selecting the best piece of content to show everyone or by appending the response desiderata in the prompt with features (e.g. fewer characters, more positive tone) common among content estimated to have high value. In my example, the former corresponds to selecting a particular article title and translating as closely as possible across languages, and the latter corresponds to asking for short titles and a more positive tone.

Surprisingly, the only approach that I have seen taken to construct content value estimates is vanilla prediction based on the GenAI content only. The predictions are generated in two steps: first, regress outcomes (e.g. time spent) onto content features (i.e. real valued embedding feature vectors or manually constructed binary indicator variables) for the GenAI content shown to the respective users; then, predict outcomes using embeddings of new candidate content.

Formally, the approach identifies content maximizing the expected outcome given G :

$$\begin{aligned}\widehat{\mathbb{E}}_n[Y \mid G] &\triangleq \underset{f \in \mathcal{F} \subseteq L_2(G)}{\operatorname{argmin}} \left\{ \widehat{\mathbb{E}}_n \left[(Y - f(G))^2 \right] \right\} \\ \widehat{g}^* &\triangleq \underset{g \in \mathcal{G}}{\operatorname{argmax}} \left\{ \widehat{\mathbb{E}}_n[Y \mid G = g] \right\},\end{aligned}$$

where Y is an outcome of interest, G is the observed GenAI content, f is square integrable with respect to $\sigma(G)$, $\widehat{\mathbb{E}}_n[V] \triangleq \frac{1}{n} \sum_{i=1}^n V_i$ for all random variables V , and \mathcal{G} is a set of GenAI content under consideration. My current impression is that this approach requires assumptions incompatible with the current paradigm (e.g. Assumption [RCT](#) or [H](#)).

Misconception: Embedding based prediction estimates value

The popularity of this vanilla prediction approach is surprising because it leverages only the GenAI content G and outcomes Y even though both depend on prompt (context) C , which is often modified based on covariates X about the user. Moreover, X is often used during tool calls that GenAI systems perform to construct responses. The set of assumptions implicit in the current paradigm motivates different estimands and estimators.

Under those assumptions, Lemma [1](#) shows how covariates X should be leveraged. Lemma [2](#) and Corollary [1](#) go a step further and show that the average observed outcome for those shown GenAI content g is generally not equal to the value of showing everyone g . In fact, Example [1](#) is a setting where they have opposite signs. Lemma [3](#) provides example settings where the two are equal, but those settings are not common in practice.

Formal Setup: Assumptions common in practice

Observed outcome Y is a sum of (1) a potential (i.e. hypothetical) outcome $Y(0)$ we would observe if an individual were shown the baseline content and (2) a potentially heterogeneous effect $\Delta(g) = Y(g) - Y(0)$ of showing GenAI content g . Observed G is assumed to be a function of only the prompt (context) C , possibly covariates X , and exogenous noise deliberately sampled to introduce stochasticity to GenAI model $G(\cdot)$. Furthermore, the particular C may be personalized based on X and depend on assignment Z in an experiment. Finally, covariates and potential outcomees are likely dependent in unknown ways. Assumptions [1](#) and [2](#) formally state the assumptions and Figure [2](#) illustrates them.

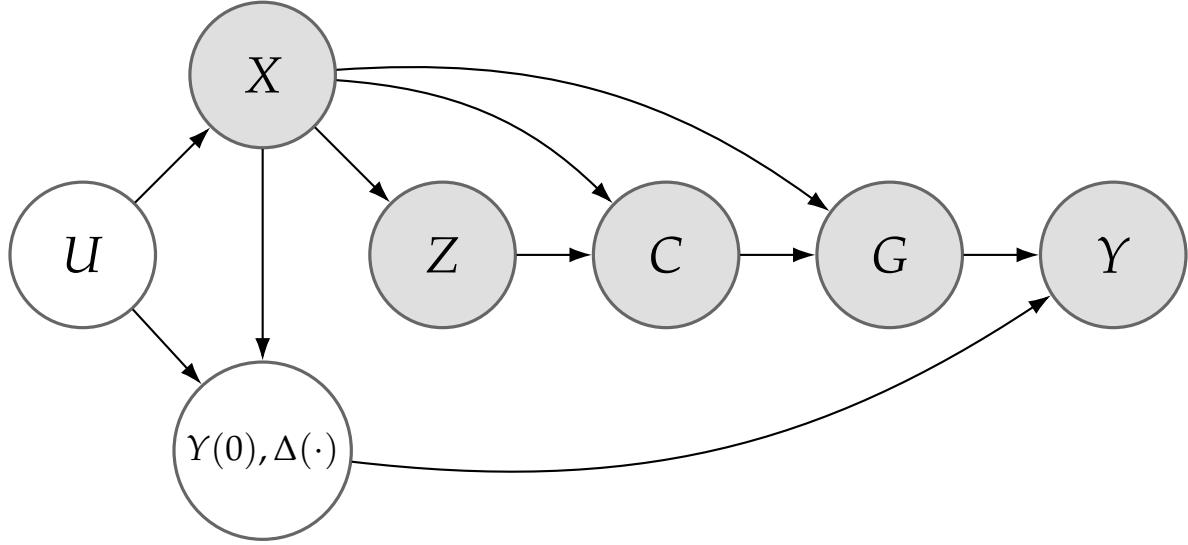


Figure 2: Directed Acyclic Graph illustrating Assumptions 1-3. Shaded circles indicate observed random variables. White circles indicate unobserved random variables. Directed edges indicate structural dependencies that are permitted though not necessary. U allows for additional possible unobservables.

Assumption 1 (Consistency).

$$\begin{aligned}
 Y(0), \Delta(\cdot), X, Z &\sim P_{Y_0, \Delta(\cdot), X} \cdot P_{Z|X} \\
 Z &= Z(X) \\
 C &= C(Z, X) \\
 G &= G(C, X) \\
 Y &= Y(G) = Y(0) + \Delta(G)
 \end{aligned}$$

where $Z(\cdot), C(\cdot), G(\cdot), Y(\cdot)$ are possibly stochastic given their respective explicit arguments above.

We do not observe or know the difference in values $\Delta(\cdot)$. We observe only covariates, assignments, prompts (contexts), GenAI content, and outcomes: X, Z, C, G, Y .

Assumption 2 (Exclusion Restriction).

$$Y \perp Z \mid X, C \tag{ER1}$$

$$Y \perp Z, C \mid X, G \tag{ER2}$$

This assumption is implied if Assumption 1 is strengthened to be read as a structural equation model illustrated in Figure 2. Settings where $\Pr\{C = c\} = 1$ or $\Pr\{Z = z\} = 1$ are special cases. That is, settings where only one prompt (context) is used.

Assumption 3 (Conditional Ignorability for Context and Content).

$$Y(G(c, X)) \perp C \mid X, Z \tag{CI1}$$

$$Y(g) \perp G \mid X, Z, C \tag{CI2}$$

Informally, Assumption [CI1](#) says that the contexts are independently assigned of the counterfactual outcome conditional on covariates, treatment assignment (if there is one). Assumption [CI2](#) is analogous. To my knowledge, both hold by construction in practice.

Assumption 4 (Conditional Overlap for Context and Content).

$$P_{C|X,Z} > 0 \quad (\text{CO1})$$

$$P_{G|X,Z,C} > 0 \quad (\text{CO2})$$

While Assumption [CO1](#) can hold by construction in experiments, and Assumption [CO2](#) is true in theory due to stochasticity introduced into $G(\cdot)$, it may be the case that the conditional probabilities of some generations are so small that they do not occur in practice.

Settings where predictions based on embeddings only should work

The following two assumptions are rarely appropriate, but if at least one of them applies, then vanilla prediction should work ([Lemma 3](#)).

Assumption 5 (Experiment for $G = g$).

$$Y(g) \perp \mathbb{I}\{G = g\} \quad \Pr\{G = g\} > 0 \quad (\text{RCT})$$

Informally, GenAI content is generated independently of the counterfactual outcome.

Assumption 6 (Homogeneous Value of g with Respect to Personalization Covariates).

$$\mathbb{E}[Y(g) | X, G = g] = \mathbb{E}[Y(g) | G = g] \quad (\text{H})$$

Informally, personalization covariates are independent (on average) of the counterfactual outcomes conditional on GenAI content. Regular conditional independence implies this.

Formal Setup: Estimands in practice

I often see two types of estimands implicit in empirical work. The first is the value of using a specific piece of GenAI content g , and the second is the value of using a specific prompt (context) c . The former is typically used to surface high value content and derive qualitative insights and features that explain value. The latter is typically used to improve prompts (e.g. system prompts) by appending desiderata for the GenAI system to satisfy (e.g. the generated content should be optimized to the persona).

Formally, the two estimands of interest are

$$g^* \triangleq \underset{g \in \mathcal{G}}{\operatorname{argmax}} \{ \mathbb{E}[Y(g)] \} \quad c^* \triangleq \underset{c \in \mathcal{C}}{\operatorname{argmax}} \{ \mathbb{E}[Y(G(c, X))] \},$$

where $\mathbb{E}[Y(g)]$ represents the value of content g and $\mathbb{E}[Y(G(c, X))]$ represents the value of prompt (context) c . To estimate these quantities, we use the above assumptions to connect these counterfactual quantities to factual ones. That is, we need the relationship between Y, G in the observed data and the hypothetical (i.e. counterfactual) $Y(g)$ so that we can perform data driven optimization.

Identification

If the context depends on covariates X , then one way to estimate the value of generations is the following three step process. First, learn a predictive model for Y using both covariates X and GenAI content G . Second, generate predictions by providing covariates X and GenAI content $g \in \mathcal{G}$ as input to the fitted model. Third, compute the average of these predictions to construct an estimate of g 's value. An analogous strategy works for estimating the value of a prompt. Both procedures are plugin estimators motivated by the following Lemma.

Lemma 1 (Identification via G-formula). *If Consistency (1), Exclusion Restriction (ER2), Conditional Ignorability (CI2), and Conditional Overlap (CO2) hold, then*

$$\mathbb{E}[Y(g)] = \mathbb{E}[\mathbb{E}[Y | X, G = g]].$$

Analogously, if (1), (ER1), (CI1), and (CO1) hold, then

$$\mathbb{E}[Y(G(c, X))] = \mathbb{E}[\mathbb{E}[Y | X, C = c]].$$

Furthermore, in general, the prediction $\mathbb{E}[Y | G = g]$ and the value $\mathbb{E}[Y(g)]$ are not equal.

Lemma 2 (Conditional Means are Not Values). *Assume (1), (ER2), (CI2), and (CO2). Then,*

$$\mathbb{E}[Y | G = g] = \mathbb{E}[Y(g)] + \text{Cov}\left(\frac{f(g | X)}{f(g)}, \mathbb{E}[Y | X, G = g]\right).$$

Thus, if the likelihood $f(g | X)$ is positively correlated with the projection of $Y(g)$ onto covariates X used for personalization, one can simultaneously have a positive prediction $\mathbb{E}[Y | G = g]$ and negative value $\mathbb{E}[Y(g)]$. The Corollary describes this formally.

Corollary 1. *If $\mathbb{E}[Y(g)] < 0$, then $\mathbb{E}[Y | G = g] > 0$ if and only if*

$$\text{Cov}(f(g | X), \mathbb{E}[Y | X, G = g]) > -f(g) \cdot \mathbb{E}[Y(g)].$$

If $\mathbb{E}[Y(g)] > 0$, $\mathbb{E}[Y | G = g] < 0$ if and only if the inequality holds with $>$ replaced with $<$.

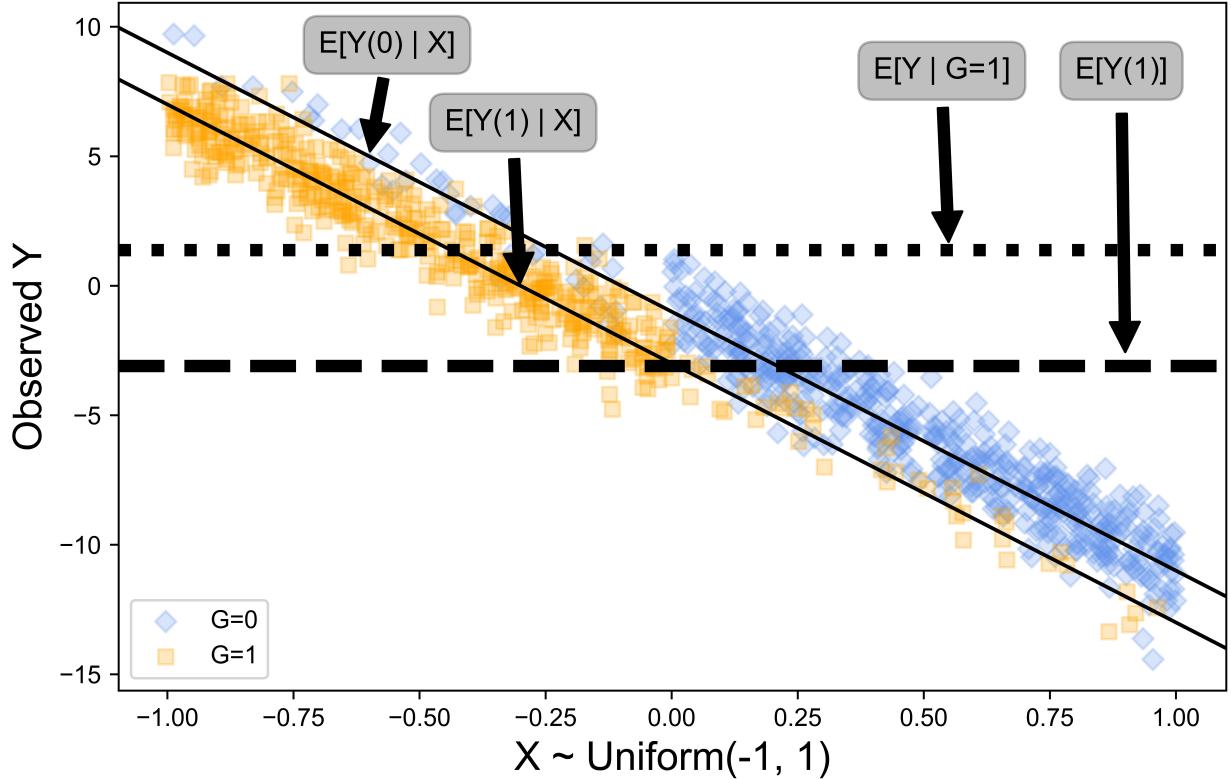


Figure 3: Illustration of the Data Generating Process in Example 1. If the decision to show content $g = 1$ to everyone depends on whether the estimated value is positive, then $\mathbb{E}[Y | G = 1]$ leads one to decide to show $g = 1$ whereas $\mathbb{E}[Y(g)]$ does not. Code to reproduce this figure is available in the Appendix.

This Corollary helps intuit sign flip examples such as Example 1, which is illustrated in Figure 3. When $G \in \{0, 1\}$ is binary, $\Delta(1)$ is negative, and $G = 1$ has greater likelihood for those with greater $Y(1)$ via G 's dependency on X , the vanilla predictions can be greater for $G = 1$ than for $G = 0$.

Example 1. Assume that $X \sim \text{Uniform}(-1, 1)$, $\eta \sim \mathcal{N}(0, 1)$, $\Pr\{G = 1 | X\} = \frac{9-8\mathbb{I}\{X \geq 0\}}{10}$, $Y(0) = -1 - 10X + \eta$, and $Y(G) = Y(0) - 2 \cdot \mathbb{I}\{G = 1\}$. Then, the following equalities hold:

$$\Pr\{X < 0\} = .5 \quad \Pr\{G = 1\} = .5 \quad \mathbb{E}[Y(1)] = -1 - 2 = -3.$$

Thus, by Bayes' Rule,

$$\Pr\{X < 0 | G = 1\} = \Pr\{G = 1 | X < 0\} \quad \Pr\{X \geq 0 | G = 1\} = \Pr\{G = 1 | X \geq 0\},$$

and by Law of Total Probability,

$$\begin{aligned}
\mathbb{E}[Y \mid G = 1] &= \mathbb{E}[Y \mid X < 0, G = 1] \cdot \Pr\{X < 0 \mid G = 1\} \\
&\quad + \mathbb{E}[Y \mid X \geq 0, G = 1] \cdot \Pr\{X \geq 0 \mid G = 1\} \\
&= \mathbb{E}[Y \mid X < 0, G = 1] \cdot \Pr\{G = 1 \mid X < 0\} \\
&\quad + \mathbb{E}[Y \mid X \geq 0, G = 1] \cdot \Pr\{G = 1 \mid X \geq 0\} \\
&= (5 - 1 - 2) \cdot \frac{9}{10} + (-5 - 1 - 2) \cdot \frac{1}{10} = 1.
\end{aligned}$$

Sign flips are alarming toy examples, but the bigger issue is that rankings over the content of \mathcal{G} can be different even when the signs agree. I suspect this is not uncommon.

That said, there exist situations where the vanilla prediction and G-formula are equal in theory. The following Lemma shows that there are at least two such settings.

Lemma 3. Assume (1), (ER2), (CI2), and (CO2). If (RCT), (H), or both hold, then

$$\mathbb{E}[Y(g)] = \mathbb{E}[Y \mid G = g].$$

Concluding Thoughts

My goals here are (1) to make clear what people are doing in both the private sector and academia and (2) propose a simple change to improve the situation. In principle, (predictive) regression of the outcome onto GenAI content (e.g. embeddings) can work when there is no personalization. Of course, this is rarely the case since prompts are often modified based on prior interactions and data collection. Moreover, GenAI systems increasingly perform tool calls that leverage auxiliary data during the response generation. For these reasons, it seems better to score content as the average (taken over users) prediction according to a model fitted on both content and covariates.

While I omitted comments on the efficiency of estimators or inference, my initial thought this is that Augmented Inverse Propensity of Treatment Weighting (AIPTW) and others could be used, but implementation will get complicated fast due to the dimensionality. For example, to estimate the generalized propensity score for vector embeddings, many additional assumptions and choices have to be made to predict densities and use them effectively for estimation.

References

Panagiotis Angelopoulos, Kevin Lee, and Sanjog Misra. Causal alignment: Augmenting language models with a/b tests. *B Tests* (April 2, 2024), 2024.

Rafael M Batista and James Ross. Words that work: Using language to generate hypotheses. Available at SSRN 4926398, 2024.

Appendix: Proofs

Proof of Lemma 1

Proof.

$$\begin{aligned}
 \mathbb{E}[Y(g)] &= \mathbb{E}[\mathbb{E}[Y(g) \mid X, Z, C, G]] && \text{(Tower Property)} \\
 &= \mathbb{E}[\mathbb{E}[Y(g) \mid X, Z, C, G = g]] && \text{(CI2, CO2)} \\
 &= \mathbb{E}[\mathbb{E}[Y(g) \mid X, G = g]] && \text{(ER2)} \\
 &= \mathbb{E}[\mathbb{E}[Y(G) \mid X, G = g]] && \text{(Conditioning event)} \\
 &= \mathbb{E}[\mathbb{E}[Y \mid X, G = g]] && \text{(1)}
 \end{aligned}$$

$$\begin{aligned}
 \mathbb{E}[Y(G(c, X))] &= \mathbb{E}[\mathbb{E}[Y(G(c, X)) \mid X, Z, C]] && \text{(Tower Property)} \\
 &= \mathbb{E}[\mathbb{E}[Y(G(c, X)) \mid X, Z, C = c]] && \text{(CI1, CO1)} \\
 &= \mathbb{E}[\mathbb{E}[Y(G(C, X)) \mid X, Z, C = c]] && \text{(Conditioning event)} \\
 &= \mathbb{E}[\mathbb{E}[Y(G) \mid X, Z, C = c]] && \text{(1)} \\
 &= \mathbb{E}[\mathbb{E}[Y \mid X, Z, C = c]] && \text{(1)} \\
 &= \mathbb{E}[\mathbb{E}[Y \mid X, C = c]] && \text{(ER1)}
 \end{aligned}$$

■

Proof of Lemma 2

Proof.

$$\begin{aligned}
 \mathbb{E}[Y | G = g] &= \mathbb{E}[\mathbb{E}[Y | X, G = g] | G = g] && \text{(Tower Property)} \\
 &= \int_X \mathbb{E}[Y | X = x, G = g] \cdot f(x | g) \cdot dx && \text{(Definition)} \\
 &= \int_X \mathbb{E}[Y | X = x, G = g] \cdot f(x) \cdot \frac{f(g | x)}{f(g)} \cdot dx && \text{(Bayes Rule)} \\
 &= \mathbb{E}\left[\mathbb{E}[Y | X, G = g] \cdot \frac{f(g | X)}{f(g)}\right] && \text{(Definition)} \\
 &= \mathbb{E}\left[\mu_g(X) \cdot \frac{f(g | X)}{f(g)}\right] && \text{(Define } \mu_g(X) \triangleq \mathbb{E}[Y | X, G = g]) \\
 &= \mathbb{E}[\mu_g(X)] \cdot \mathbb{E}\left[\frac{f(g | X)}{f(g)}\right] + \text{Cov}\left(\frac{f(g | X)}{f(g)}, \mu_g(X)\right) && (\text{Cov}(A, B) = \mathbb{E}[AB] - \mathbb{E}[A] \cdot \mathbb{E}[B]) \\
 &\stackrel{(a)}{=} \mathbb{E}[Y(g)] \cdot \mathbb{E}\left[\frac{f(g | X)}{f(g)}\right] + \text{Cov}\left(\frac{f(g | X)}{f(g)}, \mu_g(X)\right) \\
 &= \mathbb{E}[Y(g)] + \text{Cov}\left(\frac{f(g | X)}{f(g)}, \mu_g(X)\right), && (\mathbb{E}[f(g | X)] = f(g)) \\
 &= \mathbb{E}[Y(g)] + \frac{1}{f(g)} \cdot \text{Cov}(f(g | X), \mu_g(X)), && \text{(Algebra)}
 \end{aligned}$$

where equality (a) follows because

$$\begin{aligned}
 \mathbb{E}[Y(g)] &= \mathbb{E}[\mathbb{E}[Y(g) | X, Z, C, G]] && \text{(Tower Property)} \\
 &= \mathbb{E}[\mathbb{E}[Y(g) | X, Z, C, G = g]] && \text{(CI2, CO2)} \\
 &= \mathbb{E}[\mathbb{E}[Y(G) | X, Z, C, G = g]] && \text{(Conditioning event)} \\
 &= \mathbb{E}[\mathbb{E}[Y | X, Z, C, G = g]] && (1) \\
 &= \mathbb{E}[\mathbb{E}[Y | X, G = g]] && \text{(ER2)} \\
 &= \mathbb{E}[\mu_g(X)]. && \text{(Definition of } \mu_g(X))
 \end{aligned}$$

■

Proof of Corollary 1

Proof. By Lemma 2, if $\mathbb{E}[Y | G = g] > 0$ and $\mathbb{E}[Y(g)] < 0$, then the covariance below must be positive. Moreover, its magnitude must be greater than $|\mathbb{E}[Y(g)]| = -\mathbb{E}[Y(g)]$:

$$\text{Cov}\left(\frac{f(g | X)}{f(g)}, \mu_g(X)\right) > -\mathbb{E}[Y(g)].$$

Thus,

$$\text{Cov}(f(g | X), \mu_g(X)) > -f(g) \cdot \mathbb{E}[Y(g)].$$

Analogously, if $\mathbb{E}[Y | G = g] < 0$ and $\mathbb{E}[Y(g)] > 0$, then the covariance below must be negative and less than $-\mathbb{E}[Y(g)]$. ■

Proof of Lemma 3

Proof. In the first case, we have by assumption that

$$\mathbb{E}[Y(g)] = \mathbb{E}[Y(g) | G = g]. \quad (\text{RCT})$$

In the second case, we have

$$\begin{aligned} \mathbb{E}[Y(g)] &= \mathbb{E}[\mathbb{E}[Y | X, G = g]] && (\text{Lemma 1}) \\ &= \mathbb{E}[\mathbb{E}[Y(G) | X, G = g]] && (1) \\ &= \mathbb{E}[\mathbb{E}[Y(g) | X, G = g]] && (\text{Conditioning event}) \\ &= \mathbb{E}[\mathbb{E}[Y(g) | G = g]] && (H) \\ &= \mathbb{E}[Y(g) | G = g]. && (\text{Expected value of a constant}) \end{aligned}$$

In both cases,

$$\begin{aligned} \mathbb{E}[Y(g) | G = g] &= \mathbb{E}[Y(G) | G = g] && (\text{Conditioning event}) \\ &= \mathbb{E}[Y | G = g]. && (1) \end{aligned}$$

■

Appendix: Code

```

import numpy as np
import matplotlib.pyplot as plt

np.random.seed(1234)
n = 1000
X = np.random.uniform(-1, 1, n)
G = np.random.binomial(1, (9 - 8*(X > 0)) / 10)
Y0 = -1 + -10 * X + np.random.normal(0, 1, n)
effect = -2
Y = Y0 + effect * (G == 1)
avg_obs_y = Y[G==1].mean().round(4)
avg_cf_y = Y0.mean().round(4) + effect

box_kw_args = dict(
    fontsize=12, arrowprops=dict(fc="black", ec="black", shrink=0.05),
    bbox=dict(
        boxstyle="round, pad=0.5", fc="black", ec="black", lw=1, alpha=0.25
    )
)
scatter_kw_args = {
    "G=0": dict(color="cornflowerblue", alpha=0.25, label="G=0", marker="D"),
    "G=1": dict(color="orange", alpha=0.25, label="G=1", marker="s")
}

plt.rcParams["font.family"] = "Arial"
plt.figure(figsize=(8, 5))
plt.axline((0, -1), slope=-10, color="black", alpha=1)
plt.axline((0, -3), slope=-10, color="black", alpha=1)
plt.axhline(avg_obs_y, color="black", linestyle=":", lw=5, alpha=1)
plt.axhline(avg_cf_y, color="black", linestyle="--", lw=5, alpha=1)
plt.annotate("E[Y(0) | X]", xy=(-.6, 5), xytext=(-.7, 9), **box_kw_args)
plt.annotate("E[Y(1) | X]", xy=(-.3, -.2), xytext=(-.45, 6), **box_kw_args)
plt.annotate("E[Y | G=1]", xy=(.55, avg_obs_y), xytext=(.4, 8), **box_kw_args)
plt.annotate("E[Y(1)]", xy=(.9, avg_cf_y), xytext=(.8, 8), **box_kw_args)

plt.scatter(X[G==0], Y[G==0], **scatter_kw_args["G=0"])
plt.scatter(X[G==1], Y[G==1], **scatter_kw_args["G=1"])
plt.xlabel("X ~ Uniform(-1, 1)", fontsize=16)
plt.ylabel("Observed Y", fontsize=16)
plt.legend()
plt.show()

```