Counterfactual Data-Fusion for Online Reinforcement Learners

Andrew Forney [1] **Judea Pearl** [1] **Elias Bareinboim** [2]

Abstract

The Multi-Armed Bandit problem with Unobserved Confounders (MABUC) considers decision-making settings where unmeasured variables can influence both the agent's decisions and received rewards (Bareinboim et al., 2015). Recent findings showed that unobserved confounders (UCs) pose a unique challenge to algorithms based on standard randomization (i.e., experimental data); if UCs are naively averaged out, these algorithms behave sub-optimally, possibly incurring infinite regret. In this paper, we show how counterfactual-based decision-making circumvents these problems and leads to a coherent fusion of observational and experimental data. We then demonstrate this new strategy in an enhanced Thompson Sampling bandit player, and support our findings' efficacy with extensive simulations.

1. Introduction

Active learning agents are becoming increasingly integrated into complex environments, where a number of heterogeneous sources of information are available for use (4; 6; 10). These agents have the ability to both intervene in their environments (choosing actions, receiving feedback on the quality of their choices, and then modifying future actions accordingly), as well as observe other agents interacting. With the opportunity to learn from data collected from different sources other than personal experimentation come new challenges of "transfer" in learning. In particular, agents should know that actions that are desirable for populations may not be desirable for all individuals, and as such, should be wary of how observed behavior generalizes (i.e., transfers) to them and how these observations should be combined with the agent's own experience.

[1]University of California, Los Angeles, California, USA [2]Purdue University, West Lafayette, Indiana, USA. Correspondence to: Andrew Forney <forns@cs.ucla.edu>.

Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017.

In this work, we study the conditions under which data collected under heterogeneous conditions (to be defined) can be combined by an online agent to improve performance in a reinforcement learning task. This challenge is not without precedent, as recent works have investigated dataset transportability (when source and target differ structurally), though in offline domains (3; 4). Others have studied scenarios in which agents learn from expert teachers in the inverse reinforcement learning problems (1; 8). Recent work from causal analysis has addressed data-fusion for interventional quantities in reinforcement learning tasks (18). However, this work addresses data-fusion in domains where counterfactual quantities are sought, as for personalized decision-making (2; 17).

Environments for which an agent possesses fully labelled data and a fully specified model (in which all factors relating contexts, actions, and their associated rewards are known) are trivial from a learning transfer perspective; in such scenarios, collected data is homogeneous because all factors that may introduce bias between samples can be controlled. Conversely, in this paper, we focus on the challenges that arise due to *unobserved confounders* (UCs), namely, unmeasured and unlabelled variables that influence an agent's natural action choice as well as the outcome of that action. Such factors are particularly subtle when left uncontrolled due to their invisible nature and emergence of what is known as *confounding bias* (12).

Our agent's goal is to quickly learn about its environment by consolidating data collected from observing other agents and data collected through its own experience, so UCs pose a fundamental challenge: the results from *seeing* another agent performing an action are not always qualitatively the same as *doing* the action itself. As such, throughout this paper, we will differentiate three classes of data that may be employed by an autonomous agent to inform its decision-making:

1. **Observational data** is gathered through passive examination of the actions and rewards of agents other than the actor, but for whom the actor is assumed to be exchangeable.
2. **Experimental data** is gathered through randomization (e.g., standard MAB exploration), or from a fixed, non-reactive policy.

3. **Counterfactual data** (though traditionally computed from a fully specified model or under specific empirical conditions) represents the rewards associated with actions under a particular (or "personalized") instantiation of the UCs.

In the remainder of this work, we demonstrate how these data types can be fused to facilitate learning in a variant of the Multi-Armed Bandit problem with Unobserved Confounders (MABUC), first discussed in (2). In traditional bandit instances (e.g., (13; 9; 7; 14; 5)), an agent is faced with $K \in \mathbb{N}, K \geq 2$ discrete action choices (often called "arms"), each with its own, independent, and initially unknown reward distribution. The agent's task is to maximize cumulative rewards over a series of rounds, which requires learning about the underlying reward distributions associated with each arm. In the MABUC (formalized shortly), agents are faced with the same task, except that UCs modify the agent's arm-choice predilections and payout rates at each round, and the dimensionality and functional form of the UCs are unknown.

Though the data-fusion problem is an ongoing exploration in the data sciences (4), this paper is the first to study learning techniques in MABUC settings that combine data sampled under heterogeneous data-collection modes. Specifically, our contributions are as follows:

1. Using counterfactual calculus, we formally show that counterfactual quantities can be estimated by active agents empirically (Sec. 4).
2. We demonstrate how observational, experimental, and counterfactual datasets can be combined through a heuristic for MABUC agents (Sec. 4).
3. We then develop a variant of the Thompson Sampling algorithm that implements this new heuristic, and run extensive simulations demonstrating its faster convergence rates compared to the current state-of-the-art (Sec. 5).[1]

2. The Greedy Casino Revisited

In this section, we consider an expanded version of the Greedy Casino problem introduced in (2). In its new floor's configuration, the Greedy Casino is considering four new themed slot-machines (instead of the two used in the previous version) and wishes to make them as lucrative as possible. After running a battery of preliminary tests, the casino executives discover that two traits in particular predict which of the four machines that a gambler is likely to play: whether or not the machines are all blinking (denoted $B \in \{0, 1\}$), and whether or not the gambler is drunk (denoted $D \in \{0, 1\}$). After consulting with a team of psychologists and statisticians, the casino learns that any arbitrary gambler's natural machine choice can be modeled by the structural equation (12): $X \leftarrow f_X(B, D) = B + 2 * D$ if the four machines are indexed as $X \in \{0, 1, 2, 3\}$. The casino also learns that its patrons have an equal chance of being drunk or not (i.e., $P(D = 1) = 0.5$) and decide to program their new machines to blink half of the time (i.e., $P(B = 1) = 0.5$).

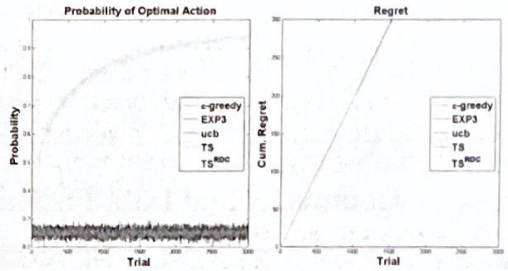

Figure 1. Plots of MAB algorithms performance vs. RDC in the Greedy Casino scenario.

To prevent casinos from exploiting their patrons, a new gambling law stipulates that all slot machines in the state must maintain a minimum 30% win rate. Wishing to leverage their new discovery about gamblers' machine choice predilections while conscious of this law, the casino implements a reactive payout strategy for their machines, which are equipped with sensors to determine if their gambler is drunk or not (assume that the sensors are perfect at making this determination). As such, the machines are programmed with the payout distribution illustrated in Table 1a.

After the launch of the new slot machines, some observant gamblers note that players appear to be winning only 20% of the time, and report their suspicions to the state gambling commission. An investigator is then sent to the casino to determine the merit of these complaints, and begins recruiting random gamblers from the casino floor to play at randomly selected machines, despite the players' natural predilections. Surprisingly, he finds that players in this experiment win 40% of the time, and declares that the casino has committed no crime. Meanwhile, the casino continues to exploit players' gambling predilections, paying them 10% less than the law-mandated minimum. Plainly, gamblers are unaware of being manipulated by the UCs B, D, and of the predatory payout policy that the casino has constructed around them. The collected data is summarized in Table 1b; the second column ($E[y_1|X]$) represents the observations drawn from the casino's floor while the third

[1] *Supplemental material.* For paper appendices and other resources, visit: https://goo.gl/MYJWbY

(a)	$D=0$		$D=1$	
$E[y_1\|X,B,D]$	$B=0$	$B=1$	$B=0$	$B=1$
$X=0$	*0.20	0.30	0.50	0.60
$X=1$	0.60	*0.20	0.30	0.50
$X=2$	0.50	0.60	*0.20	0.30
$X=3$	0.30	0.50	0.60	*0.20

(b)	$E[y_1\|X]$	$E[y_1\|do(X)]$
$X=0$	0.20	0.40
$X=1$	0.20	0.40
$X=2$	0.20	0.40
$X=3$	0.20	0.40

Table 1. (a) Payout rates decided by reactive slot machines as a function of arm choice X, sobriety D, and machine conspicuousness B. Players' natural arm choices under D, B are indicated by asterisks. (b) Payout rates according to the observational, $E[y_1|X]$, and experimental $E[y_1|do(X)]$, distributions. $Y = y_1$ represents winning.

($E[y_1|do(X)]$) represents the randomized experiment performed by the state investigator (both with large sample sizes).

In an attempt to find a better gambling strategy, an observant habitué decides to run a battery of experiments using standard MAB algorithms (e.g., ϵ-greedy, UCB, Thomson Sampling) as well as an algorithm following an approach presented in (2) known as the Regret Decision Criterion (RDC) (reviewed in the next section). Importantly, the RDC agent lacks the capability to identify and observe the UCs. The results of her experiments are depicted in Fig. 1. She notes, somewhat surprised, that all algorithms which ignore the influence of the UCs (i.e., all but RDC) perform equivalently to the randomized experiment conducted by the investigator. Noting the differences in the payout rates between the observational and experimental data, she ponders how this can be the case and how she might use these datasets to improve her gambling strategy and winnings.

3. Background

In this section, we formalize the MABUC problem in the language of Structural Causal Models (SCMs), which will allow us to articulate the notions of observational, experimental, and counterfactual distributions as well as formalize the problem of confounding due to the influence of UCs.

Each SCM M is associated with a causal diagram G and encodes a set of endogenous (or observed) variables V and exogenous (or unobserved) variables U; edges in G correspond to functional relationships relative to each endogenous variable $V_i \in V$, namely, $V_i \leftarrow f_i(PA_i, U_i)$, where $PA_i \subseteq V \setminus V_i$ and $U_i \subseteq U$; and a probability distribution over the exogenous variables $P(U = u)$.

Each M induces: (1) observational distributions $P(V = v)$, which represent the "natural" world, without external interventions; (2) a set of experimental (a.k.a. interventional) distributions $P(Y = y|do(X = x))$ for $X, Y \subseteq V$, which represent the world in which X is forced to the value x despite any causal influences that would otherwise functionally determine its value in the natural setting; and (3) a set of counterfactual distributions, defined next (12).[2]

Definition 3.1. (Counterfactual) (12) Let X and Y be two subsets of endogenous variables in V. The counterfactual sentence "Y would be y (in situation $U = u$), had X been x" is interpreted as the equality $Y_x(u) = y$, where $Y_x(u)$ encodes the solution for Y in a structural system where for every $V_i \in X$, the equation f_i is replaced with the constant x.

Note that the counterfactual expression $E[Y_x = y|X = x']$ is well-defined, even when $x \neq x'$, and is read "The expectation that $Y = y$ had X been x given that X was observed to be x'". Despite being logically valid statements in SCMs, counterfactual quantities must be estimated from either a fully specified model, or, in the absence of such, from data. In offline settings, however, counterfactual quantities are not empirically estimable (namely, when the antecedent of the counterfactual contradicts the observed value), except in some special cases ((12), Chs. 7, 9). The reason is that if we submitted the subject to condition $X = x'$, we cannot go back in time before exposure and submit the same subject to a new condition $X = x$. As is well understood in the causal inference literature, this procedure is not the same as first exposing a random unit to condition $X = x'$ since the ones who initially were inclined to act as $X = x$ are somehow different than the randomly selected subject. That said, we will show (in Sec. 4) that online learning agents possess the means to estimate counterfactuals directly.

In practice, the observational and experimental distributions can be estimated through procedures known as *random sampling* and *random experimentation*, respectively. Confounding bias emerges when UCs are present and can be seen through the difference between these two distributions, $P(Y|do(X = x)) - P(Y|X = x)$. The absence of UCs implies that $P(Y|do(X = x)) = P(Y|X = x)$, which allows random sampling (instead of a randomized experiment) to estimate the experimental distribution.

The contrast between observational and experimental data is mirrored in the distinction between *actions* (which represent reactive "choices" resulting from an agents' environments, beliefs, and other causes) and *acts* (which represent deliberate choices resulting from rational decision-making or interventions that sever the causal influences of the sys-

[2] For a comprehensive review of SCMs, we refer readers to (12).

tem (12)). To tie these concepts to the MABUC problem, one important tool introduced in (2) is known as the agent's *intent*.

Definition 3.2. (Intent) Consider a SCM M and an endogenous variable $X \in V$ that is amenable to external interventions and is (naturally) determined by the structural equation $f_x(PA_x, U_x)$, where $PA_x \subseteq V$ represents the observed parents of X, and $U_x \subseteq U$ are the UCs of X. After realization $PA_x = pa_x$ and $U_x = u_x$ (*without* any external intervention), we say that the output of the structural function given the current configuration of all UCs is the agent's intent, $I = f_x(pa_x, u_x)$.

Thus, intent can be seen as an agent's chosen *action* before its execution, which, in fact, is a proxy for any influencing UCs.[3] To ground these notions, consider again the Greedy Casino example in which the gamblers' intents are enacted on the unperturbed casino floor, but are then averaged over during the investigator's randomized study.

We can now put these observations together and explicitly define the MABUC problem:

Definition 3.3. (K-Armed Bandits with Unobserved Confounders) A K-Armed bandit problem ($K \in \mathbb{N}, K \geq 2$) with unobserved confounders (MABUC, for short) is defined as a model M with a reward distribution over $P(u)$ where, for each round $0 < t < T, t \in \mathbb{N}$:

1. *Unobserved confounders:* U_t represents the unobserved variable encoding the payout rate and unobserved influences to the propensity to choose arm x_t at round t.
2. *Intent:* $I_t \in \{i_1, ..., i_k\}$ represents the agent's intended arm choice at round t (prior to its final choice, X_t) such that $I_t = f_i(pa_{x_t}, u_t)$.
3. *Policy:* $\pi_t \in \{x_1, ..., x_k\}$ denotes the agent's decision algorithm as a function of its history (discussed shortly) and current intent, $f_\pi(h_t, i_t)$.[4]
4. *Choice:* $X_t \in \{x_1, ..., x_k\}$ denotes the agent's final arm choice that is "pulled" at round t, $x_t = f_x(\pi_t)$.
5. *Reward:* $Y_t \in \{0, 1\}$ represents the Bernoulli reward (0 for losing, 1 for winning) from choosing arm x_t under UC state u_t as decided by $y_t = f_y(x_t, u_t)$.

[3] Def. 3.2 does not require that all its influencing factors be measured or acknowledged by the agent. This definition accomodates the fact that an agent's decisions can be influenced by unknown factors, an observation that is not new to the cognitive sciences (16; 11).

[4] Using this representation, the distinction between obs. and exp. settings is made transparent – π_t copies i_t in the former, but ignores i_t and listens instead to a random device (e.g., a coin toss) in the latter.

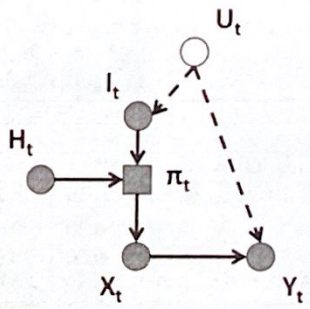

Figure 2. Model for each round of the MABUC decision process. Solid nodes denote observed variables and open nodes represent unobserved variables. The square node indicates a decision point made by the agent's strategy.

The graphical model in Fig. 2 represents a prototypical MABUC (Def. 3.3). We also add a graphical representation of the agent's history H_t, a data structure containing the agent's observations, experiments, and counterfactual experiences up to time step t. The means by which these different data-collections can be used in the agent's policy are explored at length in the next section. In summary, at every round t of MABUC, the unobserved state u_t is drawn from $P(u)$, which then decides i_t, which is then considered by the strategy π_t in concert with the game's history h_t; the strategy makes a final arm choice, which is then pulled, as represented by x_t, and the reward y_t is revealed.

Based on this definition, the regret decision criterion can be stated (2):

Definition 3.4. (Regret Decision Criterion (RDC)) (2) In a MABUC instance with arm choice X, intent $I = i$, and reward Y, agents should choose the action a that maximizes their intent-specific reward, or formally:

$$\underset{a}{\operatorname{argmax}} \; E[Y_{X=a}|X = i] \tag{1}$$

In brief, RDC prescribes that the arm $X = a$ that maximizes the expected value of reward Y having conditioned on the intended arm $X = i$ should be selected, even when $a \neq i$.

4. Fusing Datasets

Suppose our agent assumes the role of a gambler in the Greedy Casino (Sec. 2) and possesses (1) observations of arm choices and payouts from other players in the casino, (2) the randomized experimental results from the state investigator, and (3) the knowledge to use intent in its decision-making for choosing arms by the Regret Decision Criterion (RDC). In other words, the agent begins

the MABUC problem with large samples of observations ($E[Y|X]$) and experimental results ($E[Y|do(X)]$), and will maximize the counterfactual RDC ($E[Y_{X=a} = 1|X = i]$) because it recognizes the presence of UCs, viz. $E[Y|X] \neq E[Y|do(X)]$; see Table 1b). We note that the observational and experimental data available to our agent contains information about its environment, but cannot simply be incorporated into the counterfactual maximization criteria (viz. $E[Y|X] \neq E[Y|do(X)] \neq E[Y_x|x']$; see Table 1). So, the agent can choose to either discard its observations and experiments, and simply gamble by the tenets of RDC, or combine them in an informative way. This section explores the latter option.

Relating the Datasets

Note that the experimental quantity $E[Y|do(X = x)]$ can be written in counterfactual notation $E[Y_{X=x}] = E[Y_x]$, which reads as "The expected value of Y had X been x." Note also that $E[Y_x]$ can be written as a weighted average of the reward associated with arm x across all intent conditions (by the law of total probabilities), namely:

$$E[Y_x] = E[Y_x|x_1]P(x_1) + ... + E[Y_x|x_K]P(x_K) \quad (2)$$

Examining Eq. 2, we see that the equation is composed of expressions from our agent's three datasets (observational, experimental, and counterfactual). By definition, the LHS of the equation ($E[Y_x]$) is drawn from the experimental dataset. On the RHS, we have two types of quantities. Expressions of the form $E[Y_x|x']$ for which $x = x'$ are observational by the *consistency axiom* (12), because when the hypothesized and observed actions are the same, the value of y is the same (i.e., $E[Y_x|x] = E[Y|x]$). On the other hand, expressions of the form $E[Y_x|x']$ for which $x \neq x'$ are non-trivial counterfactuals, mixing observations and antecedents occurring in different worlds.

In general, evaluating counterfactuals empirically is not possible, except for some special cases, such as when the action X is binary (12). However, RDC asserts that if one preempts the agent's decision process when the intent $I = i$ is about to become a decision (X), i still encodes information about the UCs U_i (because $i = f_i(PA_x, U_x)$). This implies that randomizing within intent conditions can lead to the computation of the counterfactual given by RDC, which is a special counterfactual also called *the effect of treatment on the treated* (ETT) (12).

In order for us to exploit the properties of this equivalence to improve the performance of RDC agents in the MABUC setting, we first demonstrate that RDC indeed measures the counterfactual quantity of the ETT.

Theorem 4.1. *The counterfactual ETT is empirically estimable for arbitrary action-choice dimension (i.e., $|X| = k$ for $k \geq 2$) when agents condition on their intent $I = i$ and estimate the response Y to their final action choice $X = a$.*

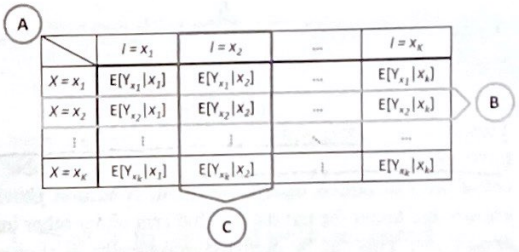

Figure 3. Counterfactual reward-history table as a cross of arm choice and intent.

For a proof of Theorem 4.1, see supplementary material, Appendix A. Because RDC is equivalently an interventional quantity, we have shown that the ETT, a counterfactual expression, can be estimated empirically through counterfactual-based randomization.[5] The main advantages of this, now proven, equivalence are threefold: (1) the empirical estimation of previously unidentifiable counterfactual quantities presents opportunities for further exploration in causal analysis, (2) the ETT's prescription for integrating experimental and observational data (see Eq. 2) permits a now interventional data-fusion strategy when such data is available, and (3) data points sampled by the agent using intent-specific decision-making are counterfactual in nature, and should therefore be added to the agent's counterfactual history. Procedures implementing RDC-type randomization should thus record intent-specific arm rewards in a table similar to Fig. 3. A second consequence of recording arm-intent-specific payouts in this fashion is that observational data may be substituted directly into cells for which the final arm choice and intent agree (see reference to consistency axiom below Eq. 2).

Strategies for Online Agents

Now that we have illustrated how the different datasets relate to our agent in the MABUC setting, we consider that the counterfactual expressions in Eq. 2 must be learned by our agent and are not known at the start of the game. Because of this finite-sample concern, we propose different learning strategies that exploit the datasets' relationship while managing the uncertainty implicit in a MAB learning scenario.

Strategy 1: Cross-Intent Learning. Consider Eq. 2 once

[5] It is understood that the ETT can be computed for binary decisions or when the backdoor criterion holds (12), but it was not believed to be estimable for arbitrary dimensions prior to RDC-randomization.

more. This holds for every arm $X = x$, which induces a system of equations as shown in Fig. 3. Consider a single cell in this system, say $E[Y_{x_r}|x_w]$, which we can solve and rewrite as:

$$E_{XInt}[Y_{x_r}|x_w] = [E[Y_{x_r}] - \sum_{i \neq w}^{K} E[Y_{x_r}|x_i]P(x_i)]/P(x_w) \tag{3}$$

This form provides a systematic way of learning about arm payouts across intent conditions, which is desirable because an arm pulled under one intent condition provides knowledge about the payouts of that arm under other intent conditions. This can be depicted graphically, as shown by row B in Fig. 3 – information about Y_{x_r} flows from intent conditions $x_i \neq x_w$ to intent x_w (a form of *information leakage*, (15)).

Strategy 2: Cross-Arm Learning. Consider any three arms, x_r, x_s, x_w such that $r \notin \{s, w\}$ and assume we are interested in estimating the value of $E[Y_{x_r}|x_w]$ (our query, for short). Considering again the equations induced by Eq. (2), we have,

$$E[Y_{x_r}] = \sum_{i}^{K} E[Y_{x_r}|x_i]P(x_i) \tag{4}$$

$$E[Y_{x_s}] = \sum_{i}^{K} E[Y_{x_s}|x_i]P(x_i) \tag{5}$$

Note that each of Eqs. (4, 5) share the same intent priors on our query intent $P(x_w)$, so we can solve for $P(x_w)$ in both equations using simple algebra, which yields,

$$\begin{aligned} P(x_w) &= \frac{E[Y_{x_r}] - \sum_{i \neq w}^{K} E[Y_{x_r}|x_i]P(x_i)}{E[Y_{x_r}|x_w]} \\ &= \frac{E[Y_{x_s}] - \sum_{i \neq w}^{K} E[Y_{x_s}|x_i]P(x_i)}{E[Y_{x_s}|x_w]} \end{aligned} \tag{6}$$

Using Eq. (6) and solving for the query in terms of our paired arm x_s, $\forall\, r \neq s$ we have

$$E[Y_{x_r}|x_w] = \frac{\left[E[Y_{x_r}] - \sum_{i \neq w}^{K} E[Y_{x_r}|x_i]P(x_i)\right] E[Y_{x_s}|x_w]}{E[Y_{x_s}] - \sum_{i \neq w}^{K} E[Y_{x_s}|x_i]P(x_i)} \tag{7}$$

Eq. (7) illustrates that any non-diagonal cell from the table in Fig. 3 can be estimated through pairwise arm comparisons with the same intent. Put differently, Eq. (7) allows our agent to estimate $E[Y_{x_r}|x_w]$ from samples in which any arm $x_s \neq x_r$ was pulled under the same intent x_w.

In practice, the online nature of the problem can make some of these pairwise computations noisy due to sampling variability when x_r is an infrequently explored arm. To obtain a more robust estimate of the target quantity, this pairwise comparison can be repeated between the query arm and all other arms with the same intent, and then pooled together. This can be seen as information about $Y_{x_r}|x_w$ flowing from arm $x_s \neq x_r$ to x_r (under intent x_w) – column C in Fig. 2(b).

One such pooling strategy is to take the *inverse-variance-weighted* average.[6] Formally, we can consider a function $E[Y_{x_r}|x_w] = h_{XArm}(x_r, x_w, x_s)$ such that h_{XArm} performs the empirical evaluation of the RHS of Eq. (7). Additionally, let $\sigma^2_{x,i} = Var_{samp}[Y_x|i]$ indicate the empirical payout variance for each arm-intent condition (as from the reward successes and failures captured by the agent in Table 3). To estimate our query from all other arms in the same intent through inverse-variance weighting, we have our now complete, second heuristic:

$$E_{XArm}[Y_{x_r}|x_w] = \frac{\sum_{i \neq r}^{K} h_{XArm}(x_r, x_w, x_i)/\sigma^2_{x_i,x_w}}{\sum_{i \neq r}^{K} 1/\sigma^2_{x_i,x_w}} \tag{8}$$

Strategy 3: The Combined Approach. The payout estimates for a MABUC algorithm using RDC (Fig. 3) can be estimated from three different sources: (1) $E_{samp}[Y_{x_r}|x_w]$, the *sample* estimates collected by the agent during the execution of the algorithm. (2) $E_{XInt}[Y_{x_r}|x_w]$, the computed estimate using *cross-intent* learning. (3) $E_{XArm}[Y_{x_r}|x_w]$, the computed estimate using *cross-arm* learning. Naturally, these three quantities can be combined to obtain a more robust and stable estimate to the target query.

We employ an inverse-variance weighting scheme so as to leverage these three estimators, and so we must formulate a metric for the payout variance associated with each strategy's computed estimate. To do so, we define an average variance for each strategy, which is the average over each sample estimate's variance (i.e., $\sigma^2_{x,i}$) used in the computation. Specifically, for the cross-arm approach (Eq. 8), we have two summations over sample payout estimates $E[Y_{x_r}|x_i]$, $E[Y_{x_s}|x_i]$ $\forall i \neq w$ which involve $2(K-1)$ terms, plus the numerator's $E[Y_{x_s}|x_w]$, giving us a total of $2(K-1)+1 = 2K-1$ variances to average. The same is true for the cross-intent apprach (Eq. 3), which involves $K-1$ sample variances to average. When estimating

[6] This strategy follows from the fact that we have Bernoulli rewards for each arm-intent condition, and as the number of samples increases for these distributions, the variance diminishes, meaning that arm-intent conditions with smaller variances are more reliable than those with larger ones.

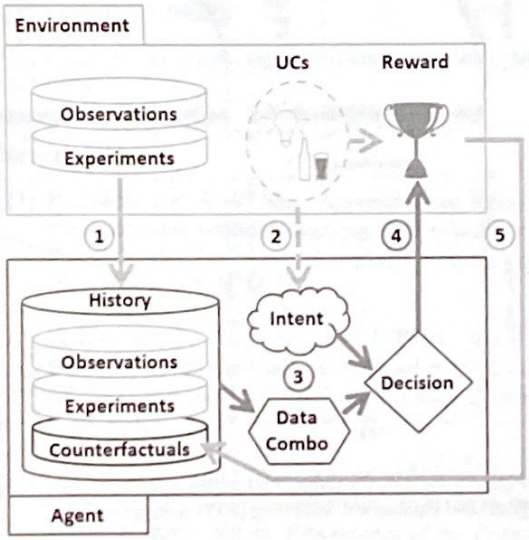

Figure 4. Illustrated data-fusion process.

$E[Y_{x_r}|x_w]$, we can write the corresponding variances:

$$\sigma^2_{XArm} = \frac{1}{2K-1}\Big[\big[\sum_{i\neq w}^{K}\sigma^2_{x_r,x_i}\big] + \big[\sum_{i\neq w}^{K}\sigma^2_{x_s,x_i}\big] + \sigma^2_{x_s,x_w}\Big]$$

$$\sigma^2_{XInt} = \frac{1}{K-1}\sum_{i\neq w}^{K}\sigma^2_{x_r,x_i}$$

Finally, to estimate $E[Y_{x_r}|x_w]$ using our combined approach, we have:

$$\alpha = E_{samp}[Y_{x_r}|x_w]/\sigma^2_{x_r,x_w} + E_{XInt}[Y_{x_r}|x_w]/\sigma^2_{XInt} + E_{XArm}[Y_{x_r}|x_w]/\sigma^2_{XArm}$$

$$\beta = 1/\sigma^2_{x_r,x_w} + 1/\sigma^2_{XInt} + 1/\sigma^2_{XArm}$$

$$E_{combo}[Y_{x_r}|x_w] = \frac{\alpha}{\beta} \tag{9}$$

To visualize the data-fusion process discussed here, consider the diagram in Figure 4.

1. In this scenario, we consider that our agent has collected large samples of experimental and observational data from its environment (e.g., in the Greedy Casino, the agent might observe other gamblers to comprise its observational data and incorporate experimental findings from the state investigator's report).

2. Unobserved confounders are realized in the environment, though their labels and values are unknown to the agent.

3. From these UCs and any other observed features in the environment, the agent's heuristics suggest an action to take, i.e., its intent. With its intent known, the agent combines the data in its history (in this work, by the prescription of Strategy 3 above) to better inform its decision-making.

4. Based on its intent and combined history, the agent commits to a final action choice.

5. The action's response in the environment (i.e., its reward) is observed, and the collected data point is added to the agent's counterfactual dataset (as a consequence of Theorem 4.1).

5. Simulations & Results

In this section, we validate the efficacy of the strategies discussed previously through simulations. To make a fair comparison to previous MABUC bandit players, we will follow the first implementation of RDC that used Thompson Sampling (TS) as its basis, embedding the strategies described in the previous section within a TS player called (TS^{RDC*}). We note that after moving from traditional to counterfactual-based decision-making we moved from optimal arm-choice nonconvergence to convergence in the MABUC setting (e.g., Fig. 1); now, the goal is to accelerate convergence.

In brief, TS^{RDC*} agents perform the following at each round: (1) Observe the intent i_t from the current round's realization of UCs, u_t. (2) Sample $\hat{E}_{samp}[Y_{x_r}|i_t]$ from each arm's (x_r) corresponding intent-specific beta distribution $\beta(s_{x_r,i_t}, f_{x_r,i_t})$[7] in which s_{x_r,i_t} is the number of successes (wins) and f_{x_r,i_t} is the number of failures (losses). (3) Compute each arm's i_t-specific score using the combined datasets via Strategy 3 (Eq. 9). (4) Choose the arm, x_a, with the highest score computed in previous step. (5) Observe result (win / loss) and update $\hat{E}_{samp}[Y_{x_a}|i_t]$.

Procedure. Simulations were performed on the 4-arm MABUC problem, with results averaged across $N = 1000$ Monte Carlo repetitions, each $T = 3000$ rounds in duration. To illustrate the robustness of each proposed strategy, we performed simulations spanning across a wide range of payout parameterizations (see Appendix B for a complete report of experimental results).

Compared Algorithms. Each simulation compares the performance of four variants of Thompson Sampling, described below and with the data-sets employed by each indicated in Table 2:

[7] The parameters for these distributions are decided by the agent's history (see Figure 2(b)), including contributions from observational data for cells in which action and intent agree.

Algorithm	Cf. Data	Obs. Data	Exp. Data
TS^{RDC*}	✓	✓	✓
TS^{RDC+}	✓	✓	
TS^{RDC}	✓		
TS			✓

Table 2. Data-sets employed by the compared TS variants.

1. TS is the traditional Thompson Sampling bandit algorithm that attempts to maximize the interventional quantity $E[y|do(x)]$, and does not condition on intent.

2. TS^{RDC} is TS player that uses RDC (Def. 3.4), but employs no additional observational or experimental data in its play.

3. TS^{RDC+} is the approach produced by (2), which uses RDC and employs observational data, but does not incorporate experimental data nor exploit the relationship between data types detailed in the previous section.

4. TS^{RDC*} follows the algorithm described above and uses the data-fusion strategy described in the previous section.

Evaluation. Each algorithm's performance is evaluated using two standard metrics: (1) the probability of optimal arm choice and (2) cumulative regret, both as a function of t averaged across all N Monte Carlo simulations. However, unlike in the traditional MAB literature, we compare each algorithm's choice to the optimal choice of an omniscient oracle that knows the value of any UCs in any given round of any MC repetition (indicated as $x^*_{n,t}$). Formally, for all $0 < t < T$ we evaluate (1) as $\frac{1}{N}\sum_n 1(x^*_{n,t} = x_{n,t})$ and (2) as $\frac{1}{N}\sum_n \sum_i^t E[Y_{x^*_{n,i}}|u_{n,i}] - y_{n,i}$.

Experiment 1: "Greedy Casino." The Greedy Casino parameterization, as described in Table 1, exemplifies the scenario where all arms are both observationally equivalent ($E[Y|x] = E[Y|x'], \forall x, x'$) and experimentally equivalent ($E[Y|do(x)] = E[Y|do(x')], \forall x, x'$), but distinguishable within intent conditions ($E[Y_x|x']$). In this reward parameterization, TS^{RDC*} experienced significantly less regret ($M = 42.23$) than its chief competitor, TS^{RDC+}, ($M = 65.04$), $t(1998) = 13.25, p < .001$.

Experiment 2: "Paradoxical Switching." The Paradoxical Switching parameterization (see Appendix B for parameters) exemplifies a curious scenario wherein $E[Y_{x_1}] = 0.5 > E[Y_{x'}], \forall x' \neq x_1$, but for which x_1 is the optimal arm choice in only one intent condition ($I = x_1$). Agents unempowered by RDC will face a paradox in that the arm with the highest experimental payout is not always optimal. Again, TS^{RDC*} experienced significantly less regret ($M = 36.91$) than its chief competitor, TS^{RDC+}, ($M = 64.70$), $t(1998) = 22.43, p < .001$.

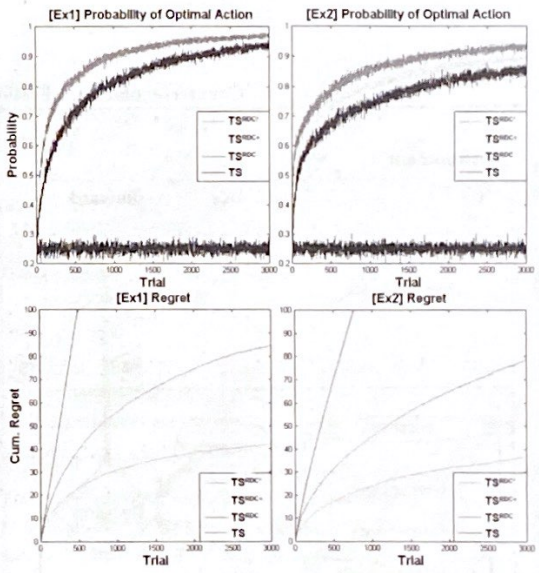

Figure 5. Plots of TS variant performances in the Greedy Casino [Ex1] and Paradoxical Switching [Ex2] scenarios.

The accelerated learning enjoyed by $RDC+$ is not localized to these parameter choices alone. In Appendix B, we show that TS^{RDC*} consistently experiences significantly less regret than its competitors across a wide range of reward parametrizations.

6. Conclusion

The present work addresses the challenges faced by online learning agents that gain access to increasingly diverse, and qualitatively different sources of information, and how these sources can be meaningfully synthesized to accelerate learning. This data-fusion problem is complicated by the presence of unobserved confounders (UCs), whose identities and influences are unknown to the modeler. In response, we present a novel method by which online agents may combine their observations, experiments, and counterfactual (i.e., personalized) experiences to more quickly learn about their environments, even in the presence of UCs. We then illustrate the efficacy of this approach in the Multi-Armed Bandit problem with Unobserved Confounders (MABUC), and demonstrate how a traditional Thompson Sampling player may be improved by its application. Simulations demonstrate that our data-fusion approach generalizes across reward parameterizations and results in significantly less regret (in some cases, as much as half) than other competitive MABUC algorithms.