Bluecore Auto-Prioritizer

09/03/2025 11:56 am EDT

Bluecore Auto-Prioritizer uses AI and reinforcement learning to determine the right frequency and priority of emails sent from Bluecore for each individual customer while maximizing overall engagement.

Are you a good fit for Auto-Prioritizer?

Think you might be a good fit for Auto-Prioritizer? There are two technical requirements:

1. You have to have at least 200k email addresses that have received a Bluecore email.
2. You have to either be using Bluecore as your full ESP, or we have to be able to capture unsubscribe events.

If you meet these requirements *and if you have a good range of campaigns running*, then we would recommend it! If you don't have a lot of campaigns sending, the model may not have enough data to work with.

Key terminology and strategies

The following document goes into more detail regarding all aspects of Bluecore Auto-Prioritizer but below is a summary of important terminology and best practices to keep in mind when using this feature:

Terminology

- **Bluecore Auto-Prioritizer Group**: Customers where email frequency is personalized by Bluecore Auto-Prioritizer.
- **Control Group**: Customers where email frequency is defined by manually set frequency caps.
- **Campaign-level and channel frequency caps**: Explained below.

Suggested Strategies

- Always include and ‘enforce’ campaign-specific frequency caps for both groups in audiences that are used with recurring campaigns.
- Apply channel frequency caps consistently across your audiences but do not enforce within the Bluecore Auto-Prioritizer group.
- Additional constraints can be applied to Bluecore Auto-Prioritizer by ‘enforcing’ additional manual frequency rules in any audience but this will also ‘narrow’ the learning potential of the model.

Getting started

This feature is automatically enabled by your Bluecore CS team and the auto-frequency and auto-prioritization is applied across **all** of your non-transactional Bluecore campaigns once active.

Initially, Bluecore Auto-Prioritizer decisioning will only be ‘exposed’ to a small percentage of your total list during the model’s learning phase. As the model begins to accumulate data and optimize over time, the percentage of your list exposed to Auto-Prioritizer decisioning will automatically increase while maintaining a control group for reporting purposes.

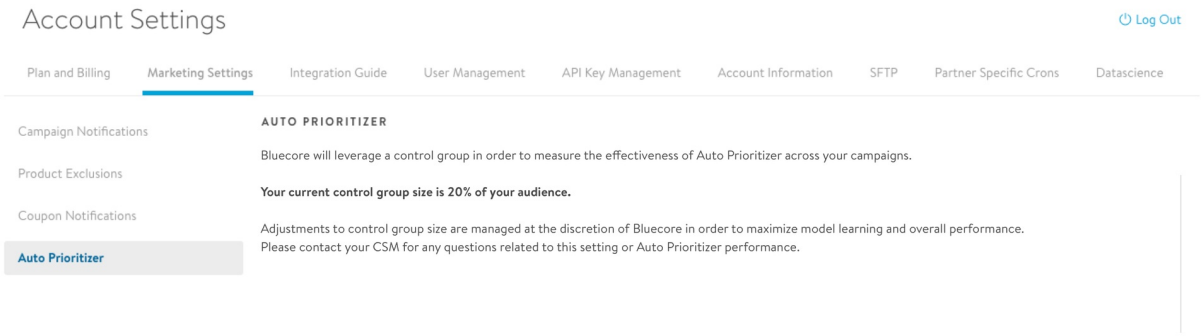


Your Bluecore CS team can provide the minimum starting *seed list* upon request.

Once fully ramped after the learning phase, a Bluecore Auto Prioritizer menu option will be visible under *Marketing Settings* that will show a read-only display of the current control group size. Access Auto-Prioritizer by following the below steps.

1. Navigate ☐ **Campaigns** > **Communicate** > **Settings**.
2. Click the **Auto-Prioritizer** tab on the marketing settings page.

Changes to this setting are automated as the model optimizes toward maximizing overall engagement while determining ideal frequency for each customer.



During the learning phase, Bluecore Auto-Prioritizer is able to, in basic terms, explore across a range of frequency settings to understand what the best choices of email frequencies are for each individual on your list is across any campaign type running with Bluecore

As it learns, Bluecore Auto-Prioritizer continuously analyzes results from these decisions and optimizes its exploration to the areas where the most engagement has occurred until ultimately finding the best choice.

After the learning phase, Bluecore Auto-Prioritizer does not stop learning but instead will continuously explore away from the best choices to either re-affirm the current decision or to find a better option if that choice has changed.

During the initial learning phase, there can be some performance fluctuations but we recommend against making significant changes to the frequency cap settings in your audience configuration during this time. Doing so can disrupt the learning phase causing delays to reaching optimal performance.

Control group

As mentioned in the *Getting Started* section, Bluecore Auto-Prioritizer will only be exposed to a minimal percentage of your list during the learning phase of the model thus the remaining portion of your list will continue to receive emails based on the same rules you've always had configured in your Bluecore audiences.

This means that the group that continues to operate under manually set rules-based frequency caps also acts as a control group to measure performance of Bluecore Auto-Prioritizer decisioning against. More details on reporting are available below.

The starting size of this group will be recommended by Bluecore and any changes to the size of this group can be made through your CSM upon request. With that said, we will always recommend that the control group persist at a statistically significant size for reporting purposes.

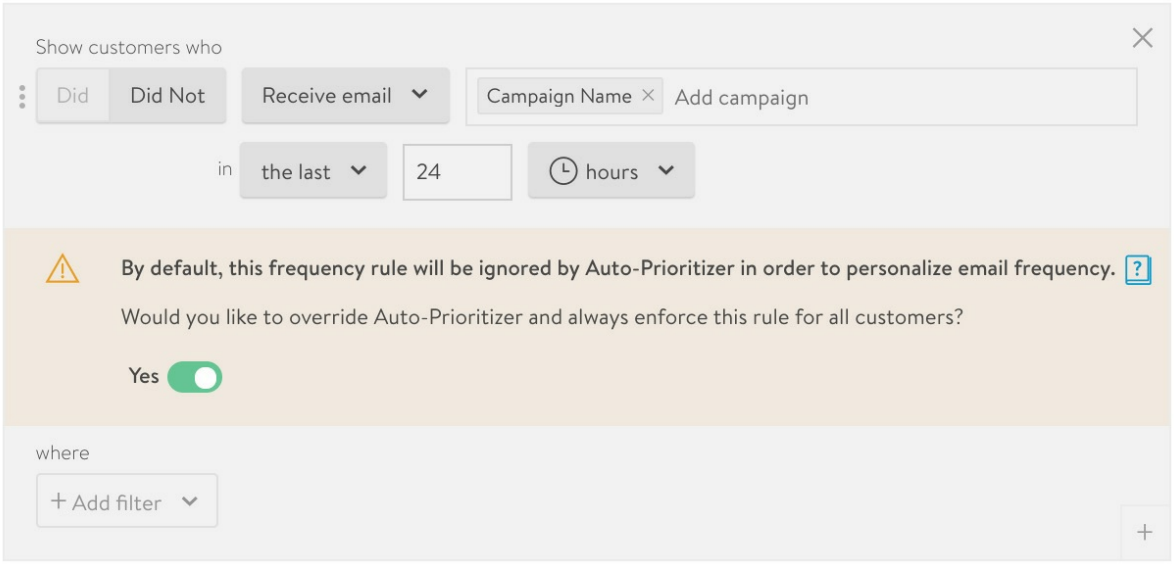
Workflows

Understand frequency caps

- Campaign-level frequency caps can be configured for campaign and campaign type suppression. These are configured in addition to channel-level caps and applied when sending a campaign.
- Channel-level frequency caps are configured within your marketing settings and are automatically applied to every campaign upon sending, unless overridden when configuring a campaign.

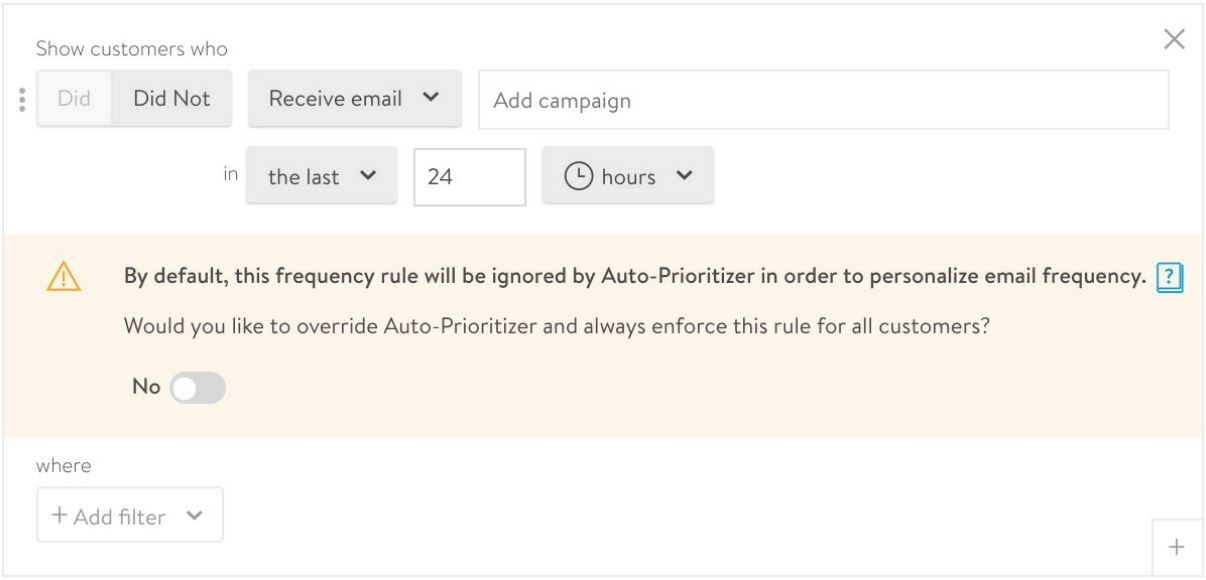
When Bluecore Auto-Prioritizer is enabled, there will be an additional option added to any step within a Bluecore Audience or Experience Designer touch that uses 'did not receive email' (i.e. a frequency cap rule) that allows you to determine when the model should always follow specific frequency constraints for all customers or bypass legacy frequency rules:

Frequency constraints can be enforced for all customers by toggling to 'Yes' as shown below:



This will ensure that this rule is never violated for any customers that qualify within the audience you are configuring.

Alternatively, leaving this toggle set to 'No' will bypass this rule for customers exposed to Auto-Prioritizer decisioning while continuing to enforce for customers in the control group. This allows for migration away from legacy audience configurations in addition to testing against any established best practices around rules-based frequency caps.

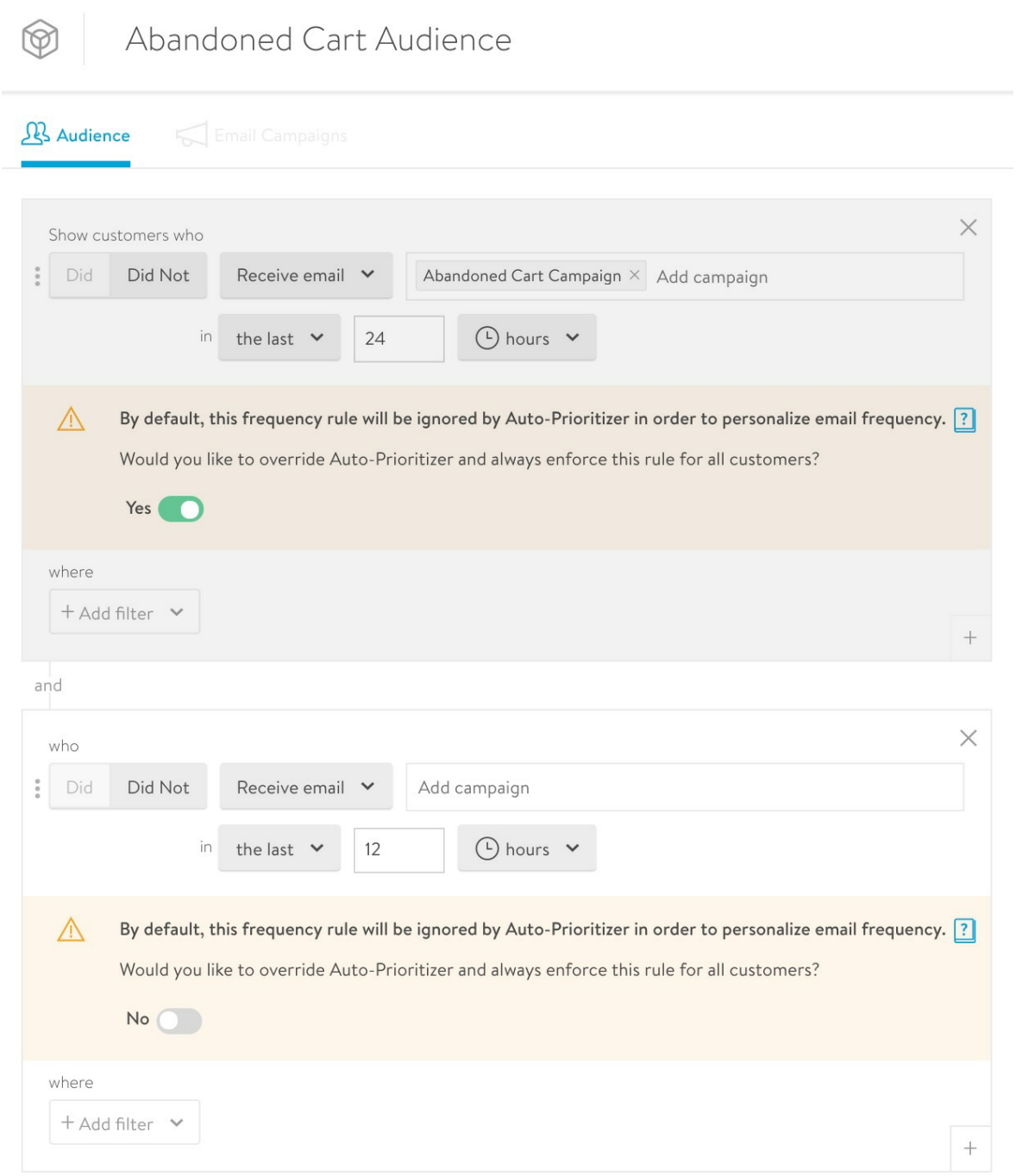


Currently, these controls allow full control for setting audience exclusions that will narrow the reach of eligible customers by effectively setting constraints (i.e. business rules) for the model to operate within.

In contrast, Bluecore Auto-Prioritizer **does not** support guaranteed inclusion of all customers within a campaign send as it may still decide that some otherwise eligible customers should not receive an email at that time.

At the time of enabling Bluecore Auto-Prioritizer , all existing campaign-specific frequency rules will be set to be enforced for all customers while all other frequency rules will default to being bypassed and only enforced within the Control group.

For example, for the campaign 'Abandoned Cart Campaign' associated with the audience 'Abandoned Cart Audience', once Bluecore Auto-Prioritizer was enabled the existing frequency caps within this audience would appear like below:



The recommended best practice when configuring new audiences once Bluecore Auto-Prioritizer has been enabled is to follow the strategy above where any business rule preference on campaign-specific frequency caps should be enforced for all customers while Bluecore Auto-Prioritizer is bypassing all other frequency caps and only applying within the control group.

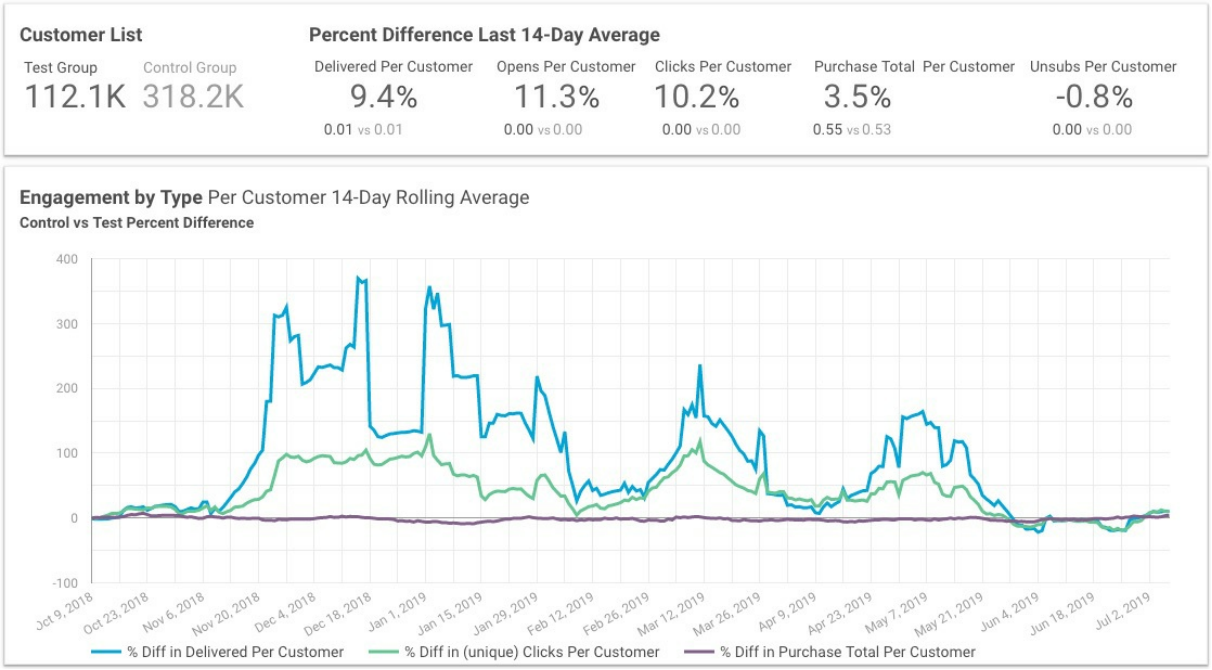
Reporting

Currently, reporting is not available within the UI but is available upon request through your CS team.

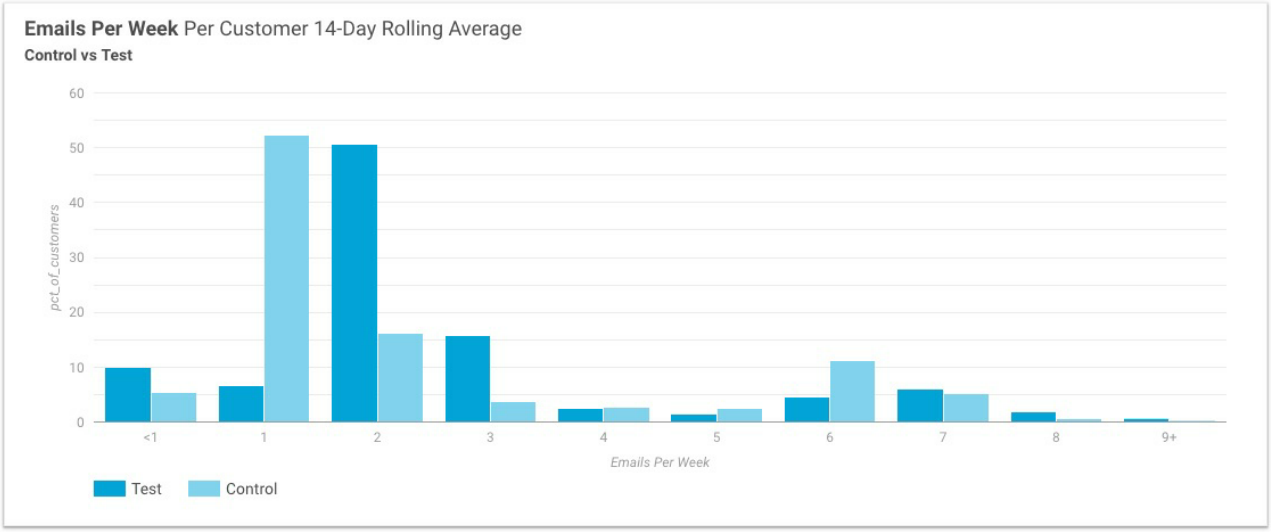
This reporting focuses on comparing Bluecore Auto-Prioritizer and Control group customers to show the percent difference in the following per customer metrics averaged over a rolling two-week timeframe:

- Delivers
- Opens
- Clicks

- Purchase Total
- Unsubscribes



In addition, there is a comparison of the average number of emails per customer per week between Bluecore Auto-Prioritizer and Control groups.



Technical overview

To find the optimal email frequency for each customer, we use a reinforcement learning approach. We model the “Bluecore universe”, containing customers and their interactions with emails and products by introducing the following concepts:

- **States**: each customer is modeled as being in a certain “state”, defined by features such as their past email opens/clicks, on-site views/purchases or other behaviors. Importantly, a customer’s state also includes whether they are unsubscribed or not.
- **Actions**: the decisions made by Auto Prioritizer can be viewed as setting an effective email frequency for each customer. Rather than just setting a channel frequency that is the same for all campaigns, we can take into account the campaign type (high/low converting, trigger vs weekly batch send, etc.) by allowing Auto Prioritizer to set different frequencies for different campaigns. We can thus think of the actions taken by the model as a high dimensional vector, with one

frequency for each type of campaign.
- **Rewards**: The “reward” for choosing the best action (vector of frequencies) for each customer is if that customer makes more purchases than he would with a suboptimal action. In the context of our reinforcement learning model, we encode the reward as the number of purchases made by a given customer on a given day.

When we first enable the model, we use a small subset of the customer base to perform random exploration, where actions are chosen at random and we observe the rewards. After that, armed with a historical dataset of states (who is this customer?), actions (what email frequency did we use?) and rewards (how many purchases did they make?), we can train a reinforcement learning algorithm to find the best action to take for each customer, in order to maximize the total reward over the lifetime of the customer. Because the optimal email frequency can change over time, we always reserve a small (and changing) cohort of customers to perform random exploration, so the model can keep making optimal decisions.

To make sure we are really maximizing for lifetime value instead of what’s best in the short term, the model does not just look at which actions cause the highest rewards, but also how the states of customers change over time, depending on the action we take. This means that the model does not always just find the best actions to make customers purchase something immediately - sometimes the better strategy can be to get customers into a better “state”, which can then lead to a more sustainable stream of purchases some time in the future. For example, we may want to first bring disengaged customers back to browsing your website and become interested in your products, and only make a purchase a little further down the road. In this way, the model is “forward-looking” instead of “greedy”.

Another benefit of the reinforcement learning approach is that the model automatically tries to minimize unsubscribes. Just like other customer behaviors, whether or not a customer is unsubscribed or not is also encoded in their state. Because unsubscribed customers generally purchase less than subscribed customers, being in the unsubscribed state is very undesirable. Once again, because we optimize for lifetime purchases, the model makes decisions so as to minimize the risk of customers going into that “unsubscribed state”, which would lead to fewer purchases in the future.