Response with technique appendix to Reviewer MnU3 of Paper: Efficient Subgame Refinement for Extensive-form Games.

Thank you for your suggestions. This technique appendix includes our response to the reviewer's concerns and provides more details that will be added to the final version's appendix.

The reviewer's questions are highlighted in blue, and our responses are in black. Any revisions or new content requested by the reviewer will be highlighted in red following the corresponding response.

A. Question 1.

I did not quite understand the paragraph at line 262 on the right side. Does the "it" in "it is equal to..." refer to the diversity metric? If so, I don't understand: having a sample with well-distributed alternate values certainly seems like it would be a good thing, but I don't see formally how it is necessary nor sufficient for diversity in the sense of Proposition 4.3. Further, I don't immediately see what the "dynamic programming" algorithm would be in this setting.

Response: Thank you for your comments.

As we stated previously:

We would like to clarify that "it" indeed refers to the diversity metric. Having a sample with well-distributed alternate values is a desirable attribute but is not sufficient on its own to satisfy the conditions of Proposition 4.3. To find a subset that fully satisfies Proposition 4.3, we would need to determine a new strategy that takes into account the influence of alternative values on strategies. This could potentially be achieved by incorporating an additional metric to measure such influence.

We will make the following improvement for the paragraph at line 262 on the right side.

Odd version. In two-player zero-sum games, it is equal to finding a subset such that the alternative value of the subset is likely to be uniformly distributed in range $[\min_{h\in S_{top}} u_2^*(\sigma_1|\mathcal{I}_2(h)), \max_{h\in S_{top}} u_2^*(\sigma_1|\mathcal{I}_2(h))]$, which can be easily obtained through techniques such as dynamic programming.

New version. In two-player zero-sum games, the metric is equal to finding a subset such that the alternative value of the subset is likely to be uniformly distributed within the range $[\min_{h\in S_{top}} u_2^*(\sigma_1|\mathcal{I}_2(h)), \max_{h\in S_{top}} u_2^*(\sigma_1|\mathcal{I}_2(h))]$, which can be easily obtained through techniques such as dynamic programming. it is important to note that well-distributed histories are necessary but not sufficient for Proposition 4.3, and the result can be improved if additional techniques, such as Abstraction, are incorporated.

It should be noted that computing Equation (5) for large infoset is difficult, therefore diversity-based generation functions approximate Equation (5) by generating histories in S'_{top}. A standard diversity-based generation function can be implemented by first generating a subset S''_{top} of S'_{top} and calculate the diverse set according to Equation (5) in S''_{top}.

Regarding the "dynamic programming" technique.

As we stated previously:

Regarding the "dynamic programming" algorithm mentioned, its purpose is to obtain well-distributed alternate values. The algorithm operates by first sorting the alternative values and subsequently breaking down the problem into sub-problems that aim to identify well-distributed alternative values among the highest n values. We will provide further details on the algorithm and include the pseudo-code in the final version of the paper to ensure clarity and comprehensibility.

We will add the following discussion and Algorithm 2 in appendix.

New version.

Implementation details of the diversity-based generation function in medium-sized games.

We first give the pseudo-code of the implemented diversity-based generative function ftailored for medium-sized games, as outlined in Algorithm 2.

Algorithm 2 Dynamic programming for diversity-based generation functions.

```
Input: set of the history index and value tuple $\mathcal{D} = \{(i, u_2^*(\sigma_1|\mathcal{I}_2(h_i)))\}$, number $n$ of genereted states.
$\mathcal{D}' \leftarrow$ sorted according to the value of $\mathcal{D}$
$m \leftarrow$ size of $\mathcal{D}'$
▷$result[i][j]$ stores the partition of $\mathcal{D}'[0:i]$ with $j$ diverse elements with $cost[i][j]$ stores the diversity measure (lower is better).
Initialize array $result[m,n] \leftarrow -1$, $cost[m,n] \leftarrow \infty$
for $i = 0$ to $m-1$ do
  $result[i][1] = 0$
  $cost[i][0] = 0$
end for
▷The transition function is $cost[i][j] \leftarrow \min_{k \in j-1,j,\dots,i-1} cost[k][j-1] + dist(\mathcal{D}'[i][1], \mathcal{D}'[last][1])$
for $i = 1$ to $m-1$ do
  for $j = 1$ to $\min(i+1, n) - 1$ do
    for $k = j-1$ to $i-1$ do
      if $j == 1$ then
        $last = 0$
      else
        $last = k$
      end if
      $total\_cost \leftarrow cost[k][j-1] + (\mathcal{D}'[last][1] - \mathcal{D}'[i][1])^2$
      if $total\_cost < cost[i][j]$ then
        $cost[i][j] = total\_cost$
        $result[i][j] = last$
      end if
    end for
  end for
end for
▷Restore the diverse set from $result$
$previous = m-1$
diversity set $\mathcal{D}'' \leftarrow \emptyset$
for $i = 0$ to $n-1$ do
  add $\mathcal{D}'[previous]$ into $\mathcal{D}''$
  $previous \leftarrow result[previous][n-i-1]$
end for
Return $\mathcal{D}''$
```

Algorithm 2 exploits the fact that the first k elements in the sorted diverse set $\mathcal{D}''$, derived from the original set $\mathcal{D}'$, constitute a diverse set $\mathcal{D}''_k$ for the subset $\mathcal{D}'_k$ of $\mathcal{D}'$, where $\mathcal{D}'_k = \{(i, u_2^*(\sigma_1|\mathcal{I}_2(h_i))) | \forall (i, u_2^*(\sigma_1|\mathcal{I}_2(h_i))) \in \mathcal{D}', \min_{v \in \mathcal{D}'_k} v[1] \leq u_2^*(\sigma_1|\mathcal{I}_2(h_i))) \leq \max_{v \in \mathcal{D}'_k} v[1]$.

B. Question 2.

(Possibly related) I view the idea of sampling with diversity as a core part of the paper, and I would like to see more elaboration about it. For example: what is the diversity function in each experiment? Further, I would like to see an ablation experiment: in each experiment, what happens if you instead, for example, set $m = k$ (so that you take your entire random sample as your subgame, ignoring the diversity function)? Does it lead to worse performance? How much worse?

Response: Thank you for your suggestion.

We would add extra experimental result as we previously responded:

Table 2. Exploitability of GS2 with different generation functions. Reward ranges in all games are normalized to lie in $[-1, 1]$. More experimental results will be presented after the experiments finish.

GAME	BLUEPRINT	DIVERSITY-BASED GENERATION WITH $k = 3$	RANDOM GENERATION WITH $k = 3$
LEDUC POKER	0.0684	0.0526	0.0563 ± 0.0033
...	...	...	...

We appreciate the reviewer's interest in the diversity-based sampling method and the request for further elaboration.

First, we would like to clarify that the diversity-based generation function used in both the medium-sized game and GuanDan is fundamentally the same. However, the implementation details differ between the two settings. In the medium-sized game, there are only two players in a zero-sum game. As such, we can select nodes with values that are uniformly distributed between the maximum and minimum values. In contrast, GuanDan features four players, and we follow Equation (5) to construct the subgame, given the sampled m histories.

Regarding the ablation study, we have conducted additional experiments to assess the impact of setting $m = k$, effectively ignoring the diversity function. Although the ablation study for GuanDan is still in progress, we can provide some preliminary results from Leduc poker for comparison. In our original paper, the results can be considered as having $m = |I|$, while the ablation experiment results correspond to $m = k$.

The following will be added to the appendix.

New version.

Ablation Study.

To further evaluate the performance of the diversity-based generation function, we conduct additional experiments of the random generation function and compare it to the diversity-based generation function.

Firstly GS2 with the random generation function is evaluated in medium-sized games. Each game is repeated for 3 times to obtain the mean and standard deviation. **More result will be added here.**

C. Question 3.

In the experiments on GuanDan, is *nested* subgame solving being performed? If so, I would like to understand what the authors do about the issue of "particle starvation"—that is, the issue that fewer and fewer nodes will be part of your subgame as the game goes on, because observations will rule out many particles. For example: are particles h being reintroduced to the game tree? If so, how are alternate values computed for such particles, and how do you know what the play probabilities are for P1 at node $h' \neq h \in \mathcal{I}_2(h)$? Do you just take these from the blueprint? For that matter, in GuanDan, how is the blueprint computed?

Response: Thank you for your valuable comments.

As we responded previously: *We appreciate the reviewer's attention to the implementation details of our approach in GuanDan and the concerns raised regarding nested subgame solving and particle starvation.*

Yes, nested subgame solving is performed in the GuanDan experiments. To address the issue of particle starvation, our GS2 algorithm introduces new h' particles when previous particles are ruled out by observations, if possible. The computation of counterfactual probabilities for the newly introduced h' particles is indeed a challenging aspect because of the lack of blueprint.

To facilitate this computation, we construct a simple blueprint strategy for our agent and its teammate, in which both players choose the action $a = \max_a Q(I, a)$ at each infoset I. Using this blueprint, we can efficiently compute the counterfactual probabilities for the newly introduced h' particles.

We will ensure that the final version includes a comprehensive explanation of these implementation details. We hope that *these revisions and clarifications will address the reviewer's concerns and further enhance the quality of our work.*

The following content will be added into the appendix.

New version.

Implementaion details of experiments in GuanDan.

1.Structure of the constructed subgame.

The construction of the gadget game used in maxmargin subgame solving (Moravcik et al., 2016) is non-trivial in GuanDan due to the increased the number of players. Although the opponents share the same payoff for a specific history, their counterfactual best response value u_i^* could be different, making the gadget game's construction infeasible. Thus, GS2 opts to construct a resolving subgame when two opponents are playing and a maxmargin gadget game when only one opponent is playing.

For the resolving subgame, alternative nodes for both opponents are added between the chance node and S'_{top} in the augmented game. Specifically, the chance player moves first, leading to an alternative node for one opponent with options to either *enter* the subgame with the chance outcome or *opt out*. If the opponent chooses to enter, the other opponent will also have the opportunity to choose.

For the maxmargin gadget game, the construction is similar with previous works (Moravcik et al., 2016) since only two players are allowing to search in this subgame.

A simple blueprint, wherein players will act according to $a = \arg\max_{a \in A(I)} Q(I, a)$, is used to compute the counterfactual probability, which is proportionally assigned to the chance outcomes.

2.Dealing with invalid particles.

In the GuanDan experiments, GS2 generates a set S_r from S_{top} with $|S_r| = 1000$ and computes S'_{top} such that $|S'_{top}| = k$ in the sampled subset. However, upon reaching a new infoset I, it is likely that some of the histories in S'_{top} are invalid due to the new observations of other players' actions in GuanDan. When $|S'_{top}| < k$, GS2 adds new histories to S_r in order to maintain $|S'_{top}| = k$ if time permits.

According to the diversity-based generation function, GS2 must regenerate m histories and then obtain the corresponding subset. However, this would lead to a complete reconstruction of the subgame tree, which contradicts the intent of nested subgame solving. In practice, GS2 retains the valid S'_{top} and identifies a new history h' in S_r that maximizes the difference between the history and the valid S'_{top}. Consequently, h' is thereby added into S'_{top} and the procedure is repeated until $|S'_{top}| = k$ or the time limit is reached. Intuitively, the new S'_{top} is well distributed while it may not be the most diverse subset of the new S_r.

Despite of the invalid histories from the observation, histories may also be unsupported by the players' strategies. This issue is exacerbated as more actions are played. Ideally, we might desire S_r to consist of histories with positive counterfactual probability by repeating generating histories and eliminate histories that are not supported. However, repeated generation would be time-consuming and undermine the applicability of the method. Fortunately, GS2 does not assume the counterfactual probability of the history and the histories with zero probability do not necessarily need to be eliminated. In spite of the presence of the zero-probability histories, only positive-probability histories are considered for the diversity-based functions.

3.Estimation of the counterfactual best response value.

When constructing the subgame, standard subgame solving methods compute the counterfactual best response value in the constructed subgame (Brown & Sandholm, 2017; Burch et al., 2014). However, the incompleteness of the opponent's infoset would lead to incorrectness for the opponent's counterfactual value. Due to the lack of techniques to calculate the exact counterfactual best response value, the value function $Q(I, a)$ is used instead. Specifically, the estimation $\tilde{u}_i^*(\sigma_1|I) = \max_a Q(I, a)$ is used for the reached infoset $I = \mathcal{I}_i(h)$.

D. Question 4.

The fact that GuanDan is a team game introduces extra concerns that, in my view, muddy the waters somewhat. For example, the notions of equilibrium for team games differ from those in non-team games–see, e.g, ”Computational Results for Extensive-Form Adversarial Team Games” (Celli and Gatti, AAAI 2018). It would be nice to have some discussion about this. Further, in practice, one would hope to allow *both* players to perform subgame solving, not just one! How much more expensive would it be to run that experiment? I would be interested to see the results of it.

Response: Thank you for your valuable comments.

As we previously responded:

We appreciate the reviewer's concerns regarding the differences in equilibrium for team games compared to non-team games and the potential benefits of allowing both players to perform subgame solving.

Indeed, the ideal approach to subgame solving in GuanDan would be to enable both players to solve and achieve an ϵ-TMECor or TME. This would potentially improve the final performance as the teammate could refine its strategy and better coordinate with our agents.

Implementing this approach, however, would require beforehand coordination, such as exchanging global seeds, to simulate the sampling process when the teammate constructs the subgame tree. This coordination would increase the complexity of subgame construction.

Additionally, allowing both players to solve the subgame would significantly increase the complexity of solving the subgame itself. The game tree would become much larger as it would need to consider the teammate's actions in the game tree and the searching conducted by the teammate, which are crucial in online solving.

For example, let $k = 10$, if 4 nodes are sampled for the second-order knowledge set for each node in current infoset, then $|S'_{top}| = 40$ for single player search and $|S'_{top}| = 400$ for two player search.

Moreover, since the two-player search is conducted, the subgame tree must be constructed based on the public state, rather than the infoset. We believe this could be problematic, as during sampling and subgame construction, no nodes may match the current private hands, which could affect the effectiveness of GS2.

The following will be added into the appendix.

New version.

Discussion of the single-player search

As discussed in Section 5, GS2 applies single-player search in GuanDan to ensure prompt decision-making. While allowing both players to search in the subgame for an ϵ-TME or ϵ-TMECor (Celli & Gatti, 2018) may potentially enhance the final payoff, it would dramatically increase the complexity of subgame construction and resolution.

In the case of TMECor, the team common knowledge subgame must be considered since the teammate lacks knowledge of our agent's private information. This necessitates GS2 to generate histories that are valid for the public information, regardless of the private information held by both players. In GuanDan, this equates to generating histories with the observed action history. Additional coordination prior to the game, such as exchanging the global seed, is required to aid the player in predicting the teammate's generated strategy. However, the solved strategy might be problematic because it may not contain either players' infosets, which implies that the constructed subgame refines the strategy at some other infosets and disregards the current one.

In the case of TME, the situation is more complex since the player would be unaware of the teammate's generation, necessitating the consideration of a full 1-team-knowledge-limited subgame. This approach is infeasible for large games.

References

Brown, N. and Sandholm, T. Safe and nested subgame solving for imperfect-information games. *Advances in neural information processing systems*, 30:689–699, 2017.

Burch, N., Johanson, M., and Bowling, M. Solving imperfect information games using decomposition. In *Twenty-eighth AAAI conference on artificial intelligence*, 2014.

Celli, A. and Gatti, N. Computational results for extensive-form adversarial team games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Moravcik, M., Schmid, M., Ha, K., Hladik, M., and Gaukrodger, S. Refining subgames in large imperfect information games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.