A Stationary distribution of the generator

Let $\mathcal{P}_{\mathcal{G}}$ denote the transition probability of the generator, where $\mathcal{P}_{\mathcal{G}}(x \mid r = x')$ denote probability of generating x condition on the prompt being x'. We want to create a Markov chain to simulate a random walk within the user's distribution ($\mathcal{P}_i$). Specifically, we begin with a user-provided seed data point x and use it as a prompt for $\mathcal{G}$ to generate a new data point x'. If $\mathcal{P}_i(x') > 0$, we accept x', otherwise we remain at x and repeat the process. We are interested in conditions under which the stationary distribution of this markov chain is $\mathcal{P}_i$

Let's assume that support of $\mathcal{P}_i$ is finite and denote it by S, let $\mathcal{P}'_{\mathcal{G}}(x'|x)$ be the transition function over $S \times S$ where $\mathcal{P}'_{\mathcal{G}}(x \mid x) = \mathcal{P}_{\mathcal{G}}(x \mid x) + \sum \mathcal{P}_{\mathcal{G}}(x' \mid x)\mathbb{I}[\mathcal{P}_i(x') = 0]$ and for $x, x' \in S$ where $x \neq x'$ we have $\mathcal{P}'_{\mathcal{G}}(x \mid x') = \mathcal{P}_{\mathcal{G}}(x \mid x')$.

If the transition graph generated by $\mathcal{P}'$ is irreducible (any state can be reached from any other state) and all its states are positive recurrent (the expected time to return to a state is finite), then the unique stationary distribution using $\mathcal{G}$ is $\mathcal{P}_i$ if the following equality holds:

$$\mathbb{E}_{x \sim \mathcal{P}_i}\left[\mathcal{P}'(x' \mid x)\right] = \mathcal{P}_i(x') \tag{1}$$

The above statement states that the probability of a data point ($\mathcal{P}_i(x)$) should be proportional to the probability of reaching to that point with the transition function of $\mathcal{P}'$.

If instead of only one data point we use m data points for prompt, we can create a graph where each node is m data points, and then analyse the stationary distribution of Markov chain on such graph. In this case, when we start from a node with m examples and prompt the language model with them in this case the probability of going to $(x', x_m, \ldots, x_2)$ from $(x_m, \ldots, x_1)$ is equal to $\mathcal{P}'_G(x' \mid r = (x_m, \ldots, x_1))$.

B Linear regression analysis

In this section, we examine the performance of CoDev in a simple linear regression scenario. Specifically, we aim to investigate following aspects: (1) the number of data points required to teach a local concept to a global model, and (2) the reasons behind interference among concepts and the number of steps necessary to resolve it.

B.1 Setup

We consider each input $x \in \mathbb{R}^d$ where only some of the data points are valid. There exist a true function $\theta^\star \in \mathbb{R}^d$ such that $y = \theta^{\star\top} x$. The support of each concept ($\mathcal{P}_i$) lays on a subspace (S_i) and all valid data points on that subspace belongs to C_i. Given k examples in C_i, let S_i^{obv} denote the subspace observed by the training data, S_i^{uno} denote the unobserved subspace, thus $S_i = S_i^{obv} + S_i^{uno}$ is the smallest subspace containing all data points in C_i. Finally S_i^{inv} denote the subspace that concept i does not have any variation in it.

As a running example, let $x \in \mathbb{R}^3$ and consider a concept where data points belonging to that concept satisfies $x_1 = x_2$. Recall that only some of the data points in this subspace are valid e.g., a point is valid if x_1 is odd thus $[1, 1, 0]$ is valid while $[2, 2, 1]$ is not. Let's assume we observed $x = [1, 1, 0]$ in that subspace with label $y = 2$. In this case we have: $S_1^{obv} = [1, 1, 0]$, $S_1^{uno} = [0, 0, 1]$ and $S_1^{inv} = [1, -1, 0]$.

We consider the overparametrized noiseless linear regression, where number of features (d) is larger than number of acquired training examples (n) (therefore, we can always interpolate all the training data) and there is no noise in observed targets. Following work of [30] which showed gradient descent on linear regression lead to min L2-norm, we assume local and global models infer the min L2 norm interpolant. As an example, for our running example the min-norm solution interpolating the concept is $\hat{\theta} = [1, 1, 0]$.

B.2 Operationalizing a concept: from disagreement to convergence

An alternate interpretation of the min-norm involves inferring the parameters by taking into account explicit constraints that require $\hat{\theta}$'s projection on S_i^{uno} and S_i^{inv} to be zero. For instance, in our current example, we can deduce the min-norm solution by solving these linear equations: ($[0,0,1]\theta = 0, [1,-1,0]\theta = 0, [1,1,0]\theta = 2$).

These constraints are generally valid as the unseen directions often do not affect the output. However, these constraints may be violated when we combine local concept data with global data, as the projection of S_i^{uno} and S_0^{obv} may not be zero. This implies that the output could change with variations in the unseen directions, leading to local models typically outperforming global models within a local concept.

To ensure both local and global models perform equally well in the local concept, we need to enforce the invariance constraints explicitly. This involves adding new data that exhibit variations in the unseen directions and demonstrating that these variations do not affect the output. Furthermore, we presume that S_i^{uno} is significantly large, making methods that attempt to examine all possible directions inefficient. Therefore, it's more advantageous to only verify directions that are affected by the merge.

Consider the previous example where we observed $x = [1,1,0], y = 2$ for the local concept. Now, imagine the we observed $x = [0,1,1], y = 2$ in global dataset. When we combine this data point with the concept data point, we get $\hat{\theta}_{\text{global}} = [\frac{2}{3}, \frac{4}{3}, \frac{2}{3}]$. This causes a disagreement in data points that vary in the $[0,0,1]$ direction within the local concept, the local model predicts 0 while the global model predicts $\frac{2}{3}$. Note that both the global and local predictions align for variations in the $[1,1,0]$ direction.

In the event of such a disagreement, we have two options: (1) The variation in this direction is indeed non-zero, suggesting the local model requires further refinement - a frequent occurrence in early stages, or (2) The variation is zero, but it needs to be specified as such; otherwise, the global model assumes other values due to its implicit bias towards generating the simplest model. Note that there is no disagreements in the common directions between S_0^{uno} and S_i^{uno} or their orthogonal subspaces.

Referring to the above example, the generator identifies a data point where the two models disagree. Let's assume this data point is $x = [0,0,1]$, where the local model predicts 0, but the global model predicts $\frac{2}{3}$. In such a case, we present this data point to the user. Let's assume user specify that the label for this data point is 0. In this case by adding this new data point the global prediction adjusts to $\hat{\theta}_{\text{global}} = [0,2,0]$.

After we learn the local concept (i.e., all the unobserved directions are indeed zero), how many of them do we need to add as explicit constraints? the following proposition shows maximum number of disagreements after learning a local concept.

Proposition 1. *If* $\mathrm{proj}_{S_i^{uno}}(\theta^\star) = 0$*, then the maximum number of disagreement between local and global models is* $\dim(\mathrm{proj}_{S_0^{obv}}(S_i^{uno} \cap (S_i^{uno} \cap S_0^{obv})^\perp))$.

Proof. The global and local models agree on all observed directions (i.e., S_i^{obv} and S_0^{obv}). However, there is a disagreement for any vector u in S_i^{uno} such that $\hat{\theta}_{\text{global}} = \mathrm{proj}_{S_0^{obv}}(\theta^\star)^\top u \neq 0$ since $\hat{\theta}_i^\top u = 0$. Let's assume we add k examples such that local and global disagree. We now prove that $k \leq \dim(\mathrm{proj}_{S_0^{obv}}(S_i^{uno} \cap (S_i^{uno} \cap S_0^{obv})^\perp))$.

For the k added examples, only consider their components in $(S_i^{uno} \cap (S_i^{uno} \cap S_0^{obv})^\perp)$ (we can remove the S_i^{obv} components by subtracting their projection on S_i^{obv} similarly remove any component in $(S_i^{uno} \cap S_0^{obv})$ by subtracting their projection in S_0^{obv}). In order to have a disagreement these data points should have non-zero projection on S_i^{obv} otherwise there will be no disagreements. As a result the maximum number of data points is $\dim(\mathrm{proj}_{S_0^{obv}}(S_i^{uno} \cap (S_i^{uno} \cap S_0^{obv})^\perp))$. □

B.3 Handling interference between concepts

In previous section, we explained why disagreement can happen between local and global model and how we can resolve the disagreements by querying user of the local concept. We bound number of disagreement with dimension of projection of S_i^{uno} on S_0^{obv}. In previous section we did not need to change S_0^{obv} but when concept j has conflicts with concept i we also add data to concept j (thus changing S_j^{obv}) which can lead to new conflicts with concept i.

Concept	Examples	Example of bugs found by CoDev
X person = not X person	How can I become a positive person? How can I become a person who is not negative?	predicts duplicate shortcut bugs: How can I become a mysterious person? / How can I become someone with no mystery? predicts non-duplicate overfit bugs: How can I become a blind person? / How can I become someone who has lost his (physical) vision?
Modifiers changes question intent	Is Mark Wright a photographer? Is Mark Wright an accredited photographer?	predicts not-duplicate shortcut bugs: Is he an artist? / Is he an artist among other people? predicts duplicate overfit bugs: Is Joe Bennett a famous court case? / Is Joe Bennett a famous American court case?

Table 5: Examples of bugs found by CoDev in the concepts introduced by CheckList, which were subsequently "debugged" using AdaTest, demonstrating that AdaTest had not yet fully operationalized these concepts.

The following proposition state that in addition to the dimension of projection of S_i^{uno} on observed subspace we also need to calculate projection on the unobserved space of different concepts as they might get added in the future. With notation of $S_{0:k}^{obv}$ denoting sum of all the S_i^{obv}, and S_{-i} denotes sum of all subspaces except i, the following proposition bounds number of times users need to add data to their concepts due to interference.

Proposition 2. *If for all i, $\mathrm{proj}_{S_i^{uno}}(\theta^\star) = 0$ then the maximum number of times that we need to handle interference is $\sum_{i=1}^{k} \dim\left(\mathrm{proj}_{S_{-i}}\left(S_i^{uno} \cap (S_i^{uno} \cap S_{0:k}^{obv})^\perp\right)\right)$.*

Proof. The proof is similar to Proposition 1. Here we need to deal with conflicts with all other topics and since it is possible that we add their unobserved subspace as well we need to compute the dimension of S_i^{uno} on the whole S_j subspace not only S_j^{obv}.

Let assume we added t example from concept i to handle interference, we now prove that $t \leq \dim(\mathrm{proj}_{S_{-i}}(S_i^{uno} \cap (S_i^{uno} \cap S_{0:k}^{obv})^\perp))$. For every data point that we add we first remove $S_{0:k}^{obv}$ components by removing its projection on $S_{0:k}^{obv}$. Now in order to have a conflict this data point should have non-zero projection on S_{-i}. As a result the maximum number of data points we can add is less or equal than $\dim(\mathrm{proj}_{S_{-i}}(S_i^{uno} \cap (S_i^{uno} \cap S_{0:k}^{obv})^\perp))$, summing over all the concept result in maximum number of interference that needs to be handled.

□

C Extra Figures

Table 5 shows some example of bugs discovered by CoDev that AdaTest was unable to find.

D Extended Version of Broader Impact and Limitations

CoDev aids in operationalizing concepts without filtering the values a user wishes the model to align with. This might inadvertently allow a malicious user to encode harmful behavior into the NLP model, a risk for which we currently have no safeguards.

CoDev's functionality greatly depends on the interconnectedness of data points within the generator (we used GPT-3 in our experiments). Consequently, in situations where we lack data for specific concepts, CoDev may not assist users in putting their concept into action, an example being the operationalization of concepts in low resource languages.

It is important to mention that we do not employ LLM for labeling tasks, hence the biases present in the Language Model (LLM) will not propagate into our model. Indeed, CoDev can be used as a tool to tackle these biases in the LLM. However, if biases exist within the LLM (e.g., sentences pertaining to certain religious contexts being closely related to those discussing violence) the user may need to engage more intensively with the system to accurately operatinalize the concept. On the other hand, a user working with a concept without any particular bias in the LLM will require less effort.

In terms of interference management, we only handled interference that arises from machine learning shortcomings and can be addressed by adding more data. However, there might be literal disagreements between users (i.e., two users assign different labels to the *same* sentence). Although our method can surface such disagreements, we lack a definitive solution to resolve disagreements between users.

Finally, our theoretical framework is limited but our goal was to gain some initial insights into why interference occurs and estimates the number of instances required to address it.

Tackling these challenges - safeguarding against malicious users, resolving literal disagreements, and conducting a more comprehensive theoretical analysis of alignment - are valuable directions for future research.