The `DispatchFromDyn` trait was implemented on something which is not a pointer or a newtype wrapper around a pointer.

Erroneous code example:

```
#![feature(dispatch_from_dyn)]
use std::ops::DispatchFromDyn;

struct WrapperExtraField<T> {
    ptr: T,
    extra_stuff: i32,
}

impl<T, U> DispatchFromDyn<WrapperExtraField<U>> for WrapperExtraField<T>
where
    T: DispatchFromDyn<U>,
{}
```

The `DispatchFromDyn` trait currently can only be implemented for builtin pointer types and structs that are newtype wrappers around them — that is, the struct must have only one field (except for `PhantomData`), and that field must itself implement `DispatchFromDyn`.

```
#![feature(dispatch_from_dyn, unsize)]
use std::{
    marker::Unsize,
    ops::DispatchFromDyn,
};

struct Ptr<T: ?Sized>(*const T);

impl<T: ?Sized, U: ?Sized> DispatchFromDyn<Ptr<U>> for Ptr<T>
where
    T: Unsize<U>,
{}
```

Another example:

```
#![feature(dispatch_from_dyn)]
use std::{
    ops::DispatchFromDyn,
    marker::PhantomData,
};

struct Wrapper<T> {
    ptr: T,
    _phantom: PhantomData<()>,
}

impl<T, U> DispatchFromDyn<Wrapper<U>> for Wrapper<T>
where
```

```
    T: DispatchFromDyn<U>,
{}
```